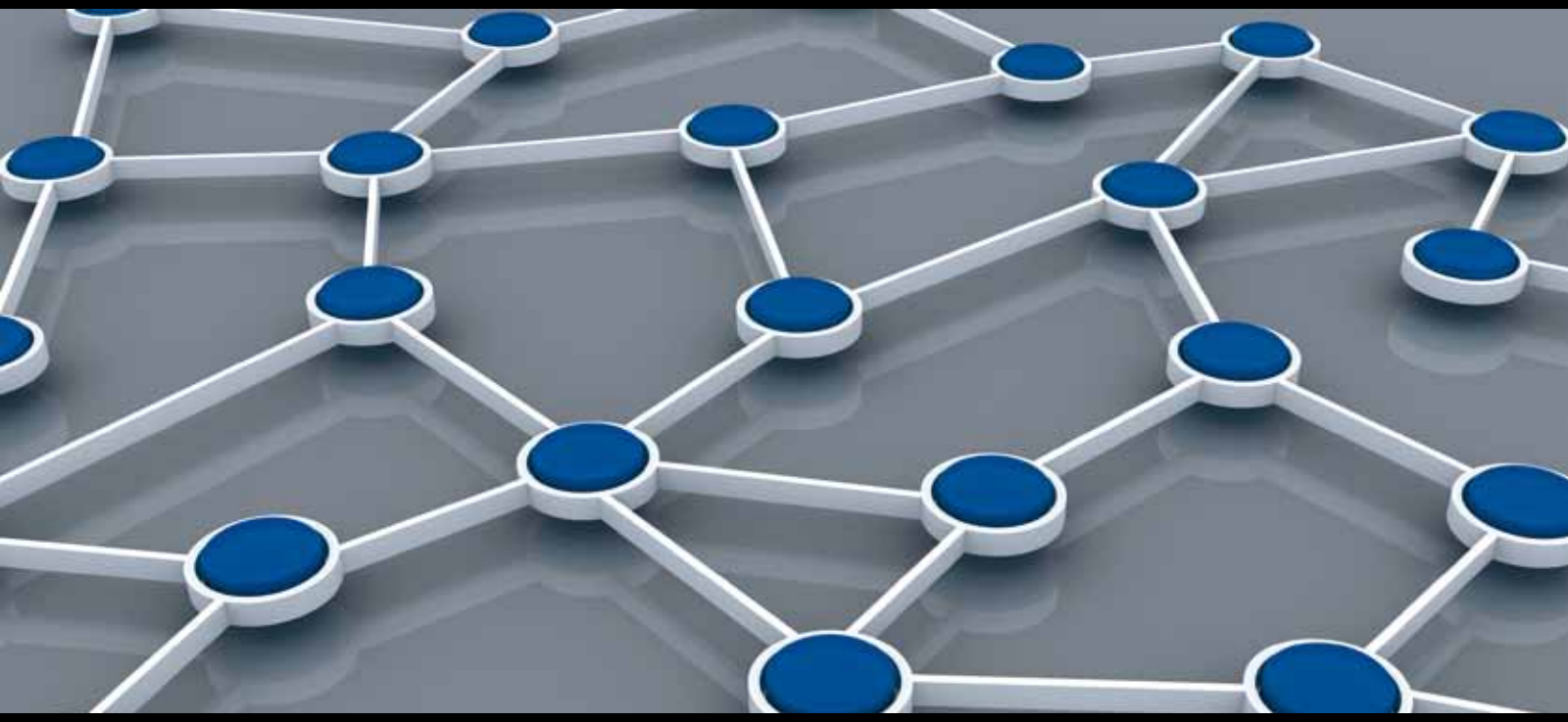
International Journal of Distributed Sensor Networks

Distributed Middleware of Large-Scale Wireless Networks

Guest Editors: Chaonong Xu, YongJun Xu, Xinrong Li, Hongsong Zhu, and Guangjie Han



Distributed Middleware of Large-Scale Wireless Networks

International Journal of Distributed Sensor Networks

Distributed Middleware of Large-Scale Wireless Networks

Guest Editors: Chaonong Xu, YongJun Xu, Xinrong Li, Hongsong Zhu, and Guangjie Han

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 431863, 2 pages
http://dx.doi.org/10.1155/2013/431863

Editorial

Distributed Middleware of Large-Scale Wireless Networks

Chaonong Xu,[1] YongJun Xu,[2] Xinrong Li,[3] Hongsong Zhu,[4] and Guangjie Han[5]

[1] *Department of Computer Science and Technology, China University of Petroleum, Beijing 102249, China*
[2] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[3] *Department of Electrical Engineering, University of North Texas, Denton, TX 76203, USA*
[4] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*
[5] *Department of Information & Communication Systems, Hohai University, Changzhou 213022, China*

Correspondence should be addressed to Chaonong Xu; xu_chaonong@ict.ac.cn

Received 22 July 2013; Accepted 22 July 2013

Large-scale wireless networks consist of a large number of nodes that communicate through wireless links. Scalability is not only one of the most important quality criterion but also a challenge to protocol design of large-scale wireless networks. Nowadays, although a great number of algorithms and protocols have been proposed, most of them are not fit for large-scale wireless networks, as they are centralized in essence, which brings formidable obstacle to scalability. Thus, developing distributed middleware is necessary for further advancing actual application of large-scale wireless networks.

This special issue is intended to provide recent advances on distributed middleware of large-scale wireless ad hoc networks and wireless sensor network. It focused on how distributed middleware has affected different aspects (protocols, algorithms, paradigm, etc.) for a large family of applications using wireless network technologies. Totally 8 papers were selected from numerous submissions after careful reviews and are divided into 3 categories.

The first category consists of 3 papers and is devoted to distributed security.

X. Zhu et al. describe a distributed traceable pseudonym management scheme in VANETs. To break risk due to centric authentication scheme, a blind signature method is adopted to achieve strict separation of issuance and tracking, which is based on the improved scheme for shared generation of RSA keys.

F. Wang et al. focus on problems of user privacy leakage and end-to-end confidentiality invasion in content-based or interest centric wireless opportunistic network. They propose a PEFKS (Public Encryption with Fuzzy Keyword Search) and CP-ABE (Cipher-text Policy Attribute Based Encryption) based distributed security scheme by refining and compromising two-pairing based encryption, searchable encryption, and attribute-based encryption. The scheme enables opportunistic forwarding according to fuzzy interests preserving full privacy of users and ensures end-to-end confidentiality with a fine-grained access control strategy in an interest-centric scenario of large-scale wireless opportunistic networks.

L. Zhu et al. focus on confidentiality-preserving data aggregation. To overcome overwhelming extra communication overhead caused by transfer list of sensors' ID, they propose a provably secure aggregation scheme PEC2P (Perturbation-based Efficient Confidentiality Preserving Protocol) that allows efficient aggregation of perturbed data without transferring any ID information.

The second category consists of 3 papers and is devoted to distributed networking.

D. Wu et al. study multimedia files transmission problem in DTN (Vehicle-Based Delay-Tolerant Network). They propose an optimal fragmentation-based multimedia transmission scheme based on P2P lookup protocol. The proposed transmission scheme enables fast and reliable multimedia file transmission in wireless mobile P2P networks over VDTNs.

N. Xia et al. study how to optimize the channel selection for wireless sniffer to maximize information collected, so as to maximize the Quality of Monitoring (QoM) for wireless networks. They propose a multiple quantum immune clone

algorithm-based solution to achieve the optimal channel allocation.

Z. Dengchang et al. study distributed time synchronization problem in large-scale wireless network. They use max-consensus to compensate for clock drift and average-consensus to compensate for clock offset. The main idea is to achieve a global synchronization just using local information.

The third category consists of 2 papers and is devoted to distributed architecture and testbed.

W. Lin et al. design an experiment platform used for teaching, research, and development of wireless sensor network. Their online Plug-Configure-Play experiment platform has four prominent strong points: in-application programming in batch, noninvasive measurement method, sensor self-awareness, and remote operation. They also develop a sensor-aware Zigbee-based smart home system prototype based on the platform.

W. Tian et al. introduce an advanced RFID middleware management system over cloud computing.

The guest editors hope that this special issue can provide a snapshot of the latest advances in distributed middleware of large-scale wireless networks and stimulate more research interest and efforts in research and development.

The guest editors would like to express their sincere gratitude to all the reviewers for their professional contributions.

Chaonong Xu
YongJun Xu
Xinrong Li
Hongsong Zhu
Guangjie Han

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 980962, 13 pages
http://dx.doi.org/10.1155/2013/980962

Research Article

An Approach to Design and Implement RFID Middleware System over Cloud Computing

Wenhong Tian, Ruini Xue, Xu Dong, and Haoyan Wang

School of Computer Science, University of Electronics and Science Technology of China, Chengdu 610054, China

Correspondence should be addressed to Wenhong Tian; tian_wenhong@uestc.edu.cn

Received 22 December 2012; Revised 11 April 2013; Accepted 20 May 2013

Academic Editor: Xinrong Li

With the development of Internet of things, the number of radio frequency identification (RFID) network readers and tags increases very fast. Large-scale application of RFID networks requires that RFID middleware system can process a large amount of data efficiently, with load-balancing, efficient redundant data elimination, and Web service capabilities so that required information can be transmitted to applications with low overhead and transparency. In view of these objectives and taking especially advantages of virtualization and transparency of Cloud computing, this paper introduces an advanced RFID middleware management system (ARMMS) over Cloud computing, which provides equipment management, RFID tag information management, application level event (ALE) management, and Web service APIs and related functions. ARMMS has different focuses than existing RFID middleware in distributed design, data filtering, integrated load balance, and Web service APIs and designs all these over Cloud. The distributed architecture can support large-scale applications, integrated load-balancing strategy guarantees stability and high performance of the system, and layered and parallel redundancy data elimination scheme makes sure that needed information is transmitted to application level with low overhead; Web service APIs support cross-platform information processing with transparency to lower level RFID hardware.

1. Introduction

Radio frequency identification (RFID) middleware plays an intermediary role between systems. Therefore, the upper-layer applications can add and delete contents or even be replaced by other software without the need to make any changes to the middleware; similarly, the underlying types of RFID readers can increase and decrease its various hardware and software operations; the upper layer does not need to make any changes. Therefore, RFID middleware can eliminate the need of many to many connections and reduces operating costs.

Figure 1 provides an overview of EPCglobal standards architecture, EPC information service (EPCIS), application level event (ALE), discovery configuration and installation (DCI), and reader management (RM) that are very much related in this paper.

As shown in Figure 2, the EPC system has several important components: readers, middleware, EPCIS, and ONS. The reader identifies and reads tags. The middleware processes all the information in tags and manages the information. The information after middleware processing is sent to the EPCIS. The EPCIS can share the information over the Internet. Finally, the object naming service (ONS) sends other tag requests to EPCIS [1]. The RFID middleware also has several fundamental functions, including data filtering, counting and aggregation of tag data, and handling the huge quantity of data generated by the RFID system [1, 2]. When facing a huge amount of data, load balance among different middleware (their hosts) is very important to keep high performance and reliability of the system [3]. Additionally, providing Web service is a way to support large-scale applications and a convenient way for management.

Figure 3 shows a possible organization of a physical RFID network.

When designing an RFID middleware solution, the following issues need to be considered [1, 2].

(a) Multiple types of hardware and vendors support: the middleware must provide a common interface to

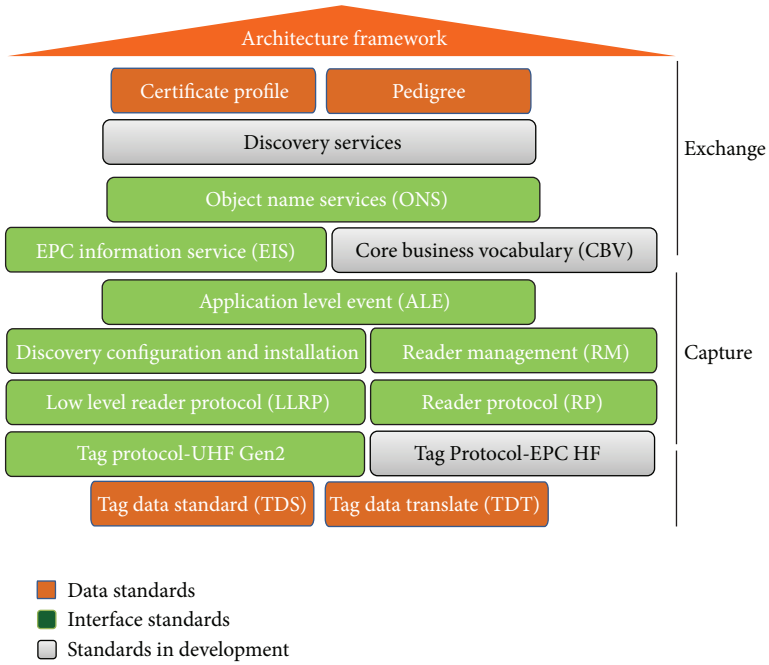


Figure 1: EPCglobal standards architecture [4].

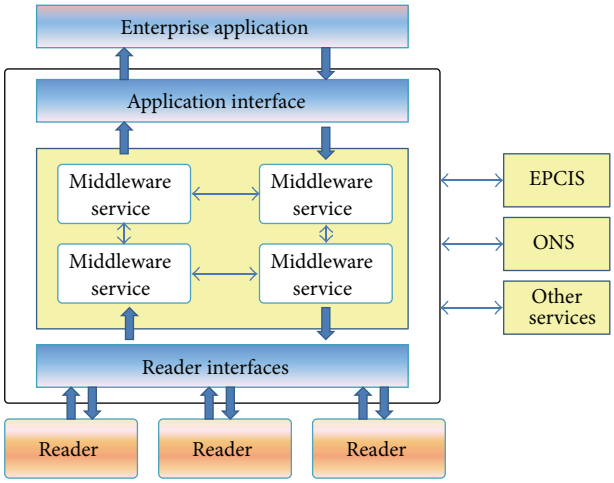


Figure 2: RFID middleware organization [1].

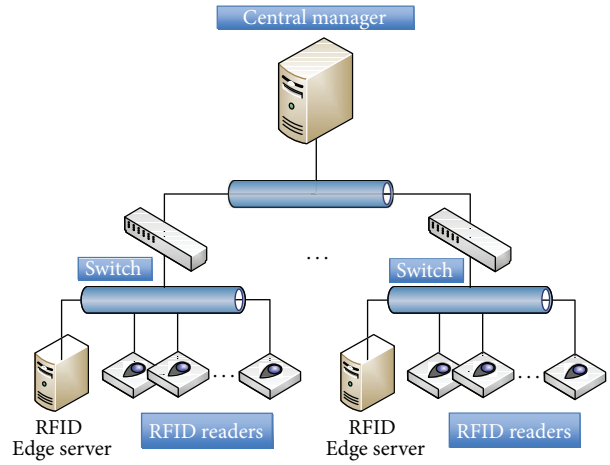


Figure 3: A networking example of RFID networks.

access different kinds of hardware by one or more vendors.

(b) Real-time handling of incoming data from the RFID readers: the middleware should handle the huge amount of data captured by the connected readers in real time without read misses.

(c) Interfacing with multiple applications: the middleware should be capable of interacting with multiple applications simultaneously, by catering to all the requirements of the applications with minimal latency.

(d) Device neutral interface to the applications: the application developer should only use the generic set of interfaces provided by the middleware independently of the type of hardware connected to the system.

(e) Scalability: the middleware design must allow easy integration of new hardware and data processing features. This needs distributed design and an open system to adapt to change.

(f) International standards compliant: the middleware design should follow international standards such as EPCglobal so that other related RFID hardware and software can also comply to the standards to make networking easier.

In view of these and taking especially advantages of virtualization and transparency of Cloud computing, this paper introduces an advanced RFID middleware management system (ARMMS) over Cloud computing, which provides equipment management, RFID tag information management, application level event (ALE) management, and Web service APIs and related functions. ARMMS has different focuses than existing RFID middleware in distributed design, data filtering, integrated load balance, and Web service APIs over Cloud.

Major contributions of this paper include

(1) proposing an RFID middleware architecture over Cloud computing including live migration;

(2) proposing a layered and parallel redundant data removal mechanism;

(3) introducing an integrated load-balance mechanism for RFID middleware;

(4) introducing Web service APIs to support cross-platform operation and information processing to improve transparency of RFID middleware to lower level hardware and software.

The rest of this work is organized as follows: Section 2 describes the related works on RFID networks and middleware design. Section 3 presents detailed features and implementation of our proposed RFID middleware system. Section 4 provides performance evaluation of proposed RFID middleware. Conclusions and future works are finally drawn in Section 5.

2. Related Research

In this section, we mainly introduce related researches on general introduction, related standards, redundancy removal, and load balance of RFID middleware design. There are several surveys in the literature [3, 5–8] that propose system taxonomies and major development in RFID middleware. General introduction of RFID technology and middleware is provided in [7–10]. Reference [5] presents a taxonomy for RFID system. There are many researches on standardization of the RFID middleware system. Massachusetts Institute of Technology (MIT) proposes EPCglobal standards. Many researchers have proposed various extensions based on the standards and implementation methods, such as in [11–13] which focus on an RFID middleware information addressing. Reference [14] is about Class 1 General 2 UHF air interface protocol. Reference [4, 15] introduces EPCglobal, EPCglobal ALE middleware design standards, ALE middleware, and messaging and device management standards.

There are many researches on RFID middleware design. Reference [16] studies RFID middleware for distributed large-scale systems. Reference [17] reviews the state-of-art RFID middleware. Reference [3] discusses the application requirements and RFID constraints for middleware. Reference [6] introduces the basics of RFID networks and middleware including Savant, WinRFID, IBM WebSphere RFID middleware, Sun Java RFID system, and FlexRFID with a focus on security, privacy, and business rules. Reference [18] presents a lightweight RFID middleware design (through temporary database implementation). Reference [19] introduces a middleware called WinRFID. Reference [20] shows the behavior and performance of message-oriented middleware system.

Some researches on redundant data removal are as follows. Reference [21] introduces a mechanism filtering redundant data in large-scale applications of RFID inventory. Reference [22] proposes an efficient filtering algorithm CLIF, for the detection and elimination of redundant data within a network. Reference [23] defines the EPC global standard of RFID tag data and proposed in-network phased filtering mechanisms (INPFM). Reference [24] proposes an adaptive RFID data-smoothing filter SMURF. Reference [25] introduces tuning approach in data filtering to reduce energy consumption in wireless sensor networks. Reference [26] presents an energy-efficient in-network RFID data filtering scheme in wireless sensor networks that has better performance regarding computational and communication costs than INPFM and CLIF. Tian et al. [27] present a parallel method to redundancy data elimination in RFID networks by applying MapReduce and Hadoop cluster in Cloud computing. INPFM and CLIF eliminate duplicated data during tags transmission phase. INPFM filters duplicated data at k hop distance. As pointed out in [26], several problems still exist by applying INPFM or CLIF. Problems include that computation cost is high, duplicate transmission cost is high, and inducing large delays when the total number of tags increase. EIFS [26] is claimed to have better performance than INPFM and CLIF regarding computational and communication cost; however, it can just eliminate about 80% of all redundancy, which may still be a problem for upper layer applications. Another issue for EIFS is that it just simulated a small number of tags (max 500 tags). In this paper, we propose a parallel method for redundancy elimination, which can remove 100% redundant data for large scale of data up to 100 million.

Many researches on load balance of RFID middleware have been conducted. Reference [28] introduces a number of RFID tag-based middleware load-balancing strategies which are good for all middleware and their hosting servers which have same configuration (homogenous case). Reference [29] introduces a distributed and agent-based design of load balance system for RFID middleware, which applies mobile and stationary agents to gather information and execute load balancing. Reference [1] proposes a grid-based load-balancing mechanism for RFID middleware applications, which incorporates functional modules buffer management and load balancing management over a grid networking platform, to buffer the read data and share the middleware loading, and compared the processing time and the packet lost ratio to the existing methods. Reference [30] introduces a centralized method for load-balancing method using connection pool in RFID middleware. Reference [31] introduces a dynamic load balancing approach based on the standard RFID middleware architecture by considering interdependencies due to RFID readers in contrast to most of the existing approaches where independencies are assumed among jobs. Reference [32] introduces a framework to

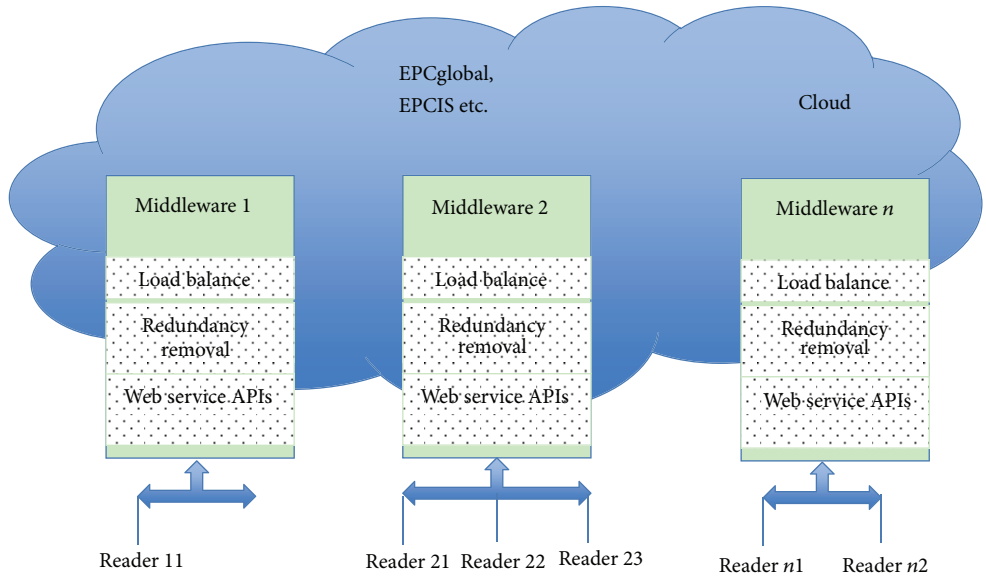


Figure 4: Proposed ARMMS middleware architecture.

dynamic and integrated load balancing for RFID middleware by considering heterogeneous configuration and multiple factors such as CPU and memory of middleware hosting servers.

3. Architecture and Features of Proposed RFID Middleware

Figure 4 shows proposed ARMMS middleware architecture based on Cloud computing. The ARMMS middleware system can be built based on Cloud computing. Especially for the load balance, hosting servers can be set up on virtual machines which can bring benefits to allocation, migration, and security; for redundancy removal, layered mechanism and parallel processing of MapReduce in Cloud computing can be applied; for Web service (APIs), dynamic web serving based on elastic Cloud computing can be applied. We will introduce these features in detail in the following section.

ARMMS has a very clear hierarchy from bottom to up as shown in Figure 5: Reader API Interface, Edge servers, RFID middleware manager/ALE server, middleware API, and RFID applications. In the following, these five layers are introduced briefly.

RFID Reader APIs. Different RFID readers should provide related APIs to middleware system to access so that functions such as equipment management, data filtering and aggregation, snd load-balance, can be conducted.

Edge Server. The main function of an Edge server is to run a single middleware software, directly manage readers by calling their APIs, acquire tag data sent by readers, and execute preliminary tag filtering algorithm to filter out redundant tag data generated by a reader.

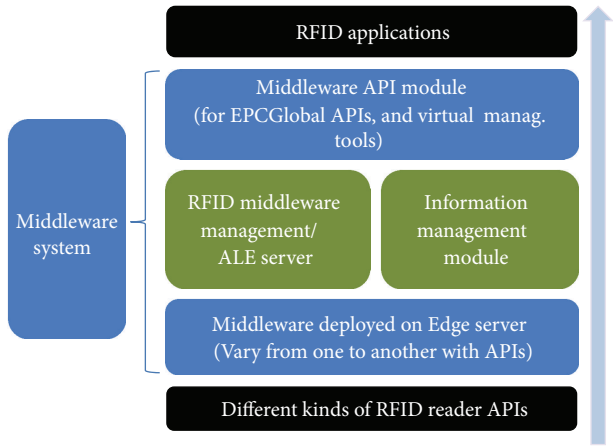


Figure 5: The structure of ARMMS.

Middleware Manager/ALE Server. Middleware manager/ALE server is primarily responsible for a single middleware software management on an Edge server according to the host load.

Information Management Module. Providing related distributed messaging queues for several applications to achieve asynchronous communication. The application may not be only on one machine but can be on different machines in a local area network. Figure 6 shows messaging mechanism in distributed ARMMS.

Middleware API Module. According to EPCglobal APIs, this module provides basic API operations including ALE events manipulation and information checking operations. Meanwhile, the visual management tools may also include a hierarchy of the visual management tools, mainly for

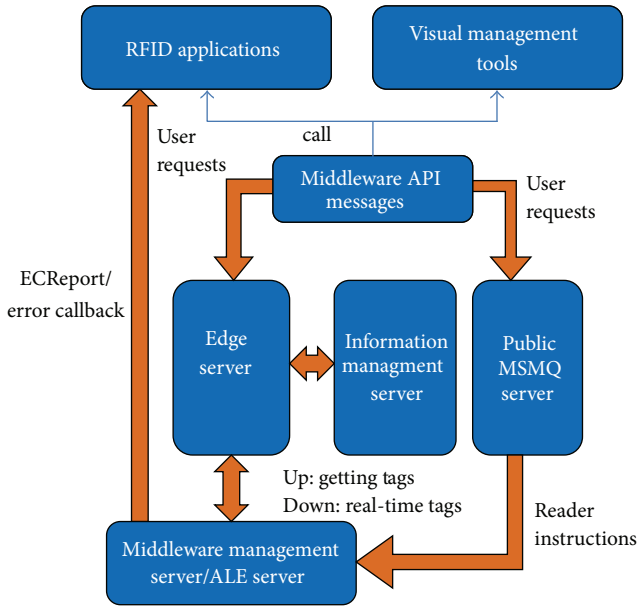


Figure 6: Messaging in distributed ARMMS.

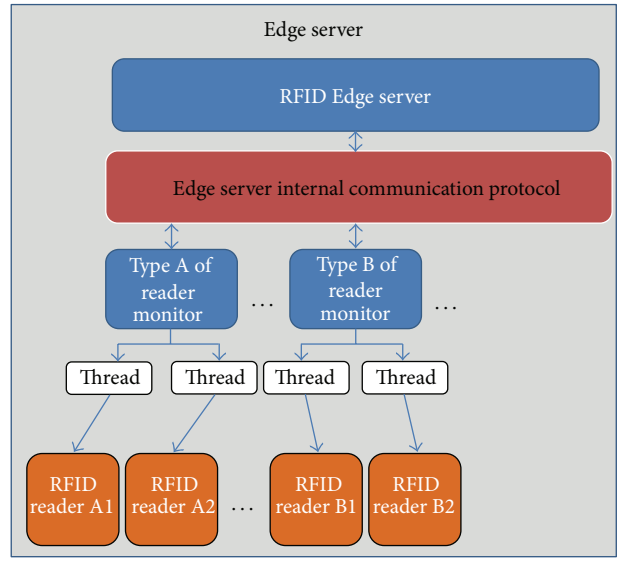


Figure 7: Structure of proposed Edge server.

providing more convenient modification and configuration of the middleware.

RFID Applications. Upper layer applications can interface with ARMMS middleware system to provide different service.

3.1. Edge Server Design. Edge server is made up of two types of backend applications, RFID Edge server and Monitor process. RFIDEdgeserver is responsible for providing message queues and collecting and distributing messages. An Edge server runs only one RFID Edge server process. Monitor backend process is the only object calls a specific RFID device API, one per type of RFID device. It uses an internal communication protocol to communicate with RFID Edge server. For a specific type of RFID device, a monitor process should be implemented. So, if we want to extend this middleware system to support a new type of RFID device, we should implement a new monitor process with the new device's API and internal communication protocol. Figure 7 shows the structure of proposed Edge server.

The main function of the monitor backend process includes the following.

(1) Calling the corresponding RFID devices' API and obtaining the tag data in the reading range.

(2) Monitoring the RFID devices connected with Edge server in real time and sending the error reports to Center Node (management center).

(3) Sending TagDetected message to the RFID Edge server process when a tag data is arrived.

RFID devices from different manufactures apply different API calling mechanisms. If middleware system calls the API directly, the scaling ability will undoubtedly become very poor. Applying support for new type of RFID device will lead to recompiling the source code. Despite operation mechanisms for reading and writing varying from one type of device to another, it is easy to find that they obtain tag data in a similar way. Edge server real-time tag algorithm is designed for the hardware abstract layer and implementing first level (reader level) redundant data elimination. We allocate a real-time tag cache for each RFID device object in the monitor thread, the real-time algorithm is responsible for assuring the tag data in the cache is in the device reading/writing range.

3.2. ALE Server Design. The requirements of central management node are in ECSpec and formatted as xml document. ALE server will transform the xml document to ECSpec object in memory when it gets the ALE requirement from central management node and initial an ECSpecUnit object with data in ECSpec to handle the requirement. The ECBoundarySpec part in ECSpec defines the ALE executing arguments, like recycle reading time, reading trigger, and so forth. The ECReportSpec part defines the report feeding back, like tag filtering pattern, grouping pattern, and so forth. So, the ECSpec is the only input in the ALE middleware system, it includes all the arguments, which are needed in ALE executing mechanism. There is a timer in the ECSpecUnit object. ALE mechanism is all depending on this clock. The ALE server flowchart is provided in Figure 8.

3.3. Central Management Node. The center node includes backend process and soap process. Soap process is responsible for communication with the RFID application who calling the middleware API. The relevant standard is "ale_1_1_1-standard-XML and SOAP bindings-20090313." Backend process is responsible for the load balance of the RFID device connection to Edge server. The Edge server load balance is shown in Figure 9 and will be explained in detail in Section 3.5.

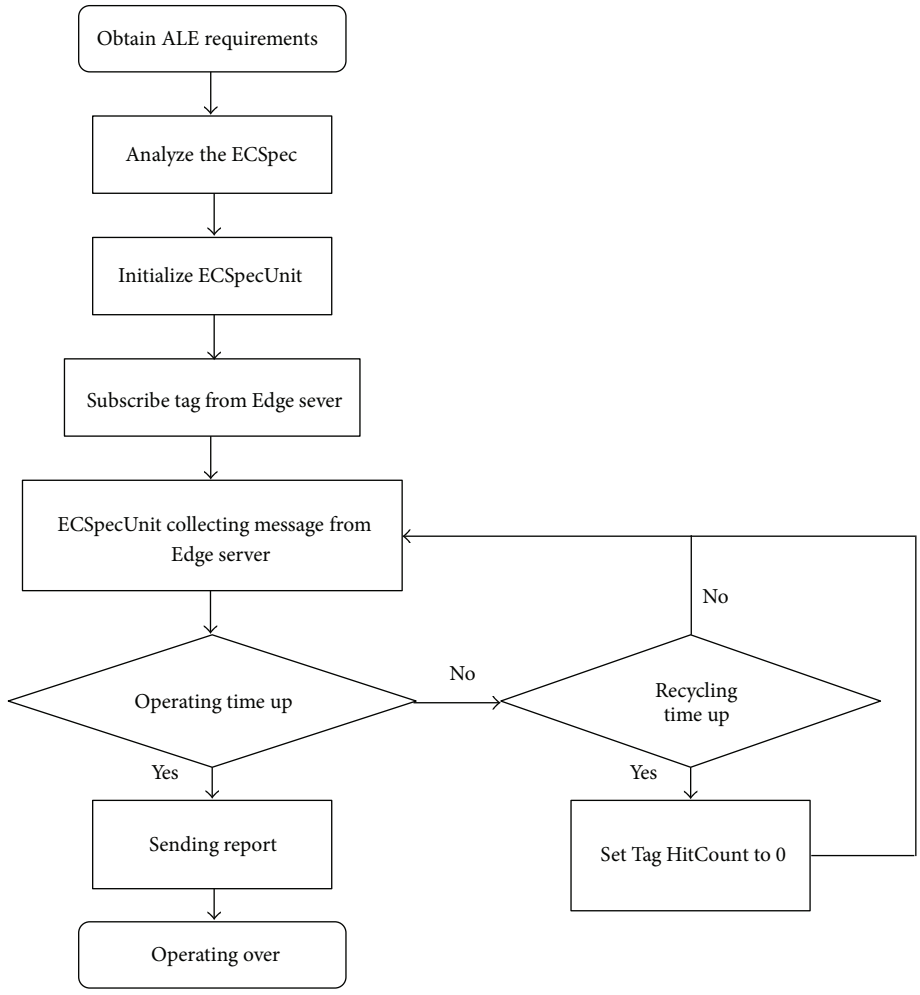


Figure 8: Flowchart of ALE server process executing.

3.4. Layered Redundant Elimination Mechanism. Figure 10 shows the data redundancy in RFID system, which can be divided into the following categories.

3.4.1. Redundancy within a Reader. When the reader gets an accurate reading of data to obtain object information on a tag, multiple times of same tag information can be read within a short period, so that redundant tags are generated.

3.4.2. Redundancy between Readers. Many readers are often densely installed to cover the entire region, and these readers' reading range may overlap with nearby readers (as shown in Figure 10); when two or more adjacent readers read the same tag information in the overlap region, redundant (repeating) tags information can be obtained.

3.4.3. Redundancy between Different Logical Groups. Different logical groups can be formed by functionality or location differences. When two or more adjacent logical groups read the same tag information in the overlap region, redundant (repeating) tags information can be obtained. Redundancy of a reader and between readers cannot be completely removed because of locality view, this is shown as the redundancy elimination percentage in [22, 23]. It needs a global elimination by collecting all tag data after reader layer and logical layer removal. To the best of our knowledge, there is no approach proposed using MapReduce [33] for redundancy elimination in RFID networks. This paper proposes a parallel MapReduce method to the redundancy data elimination. MapReduce is a programming model and can be applied to a cluster that consists of a large number of machines. Parallel MapReduce method can greatly increase the speed and efficiency of eliminating redundancy. Figure 11 presents our proposed MapReduce flowchart for redundancy elimination.

3.5. Dynamic and Integrated Load-Balancing Mechanism. Workloads of RFID middleware can change from location to location and can vary at different times. There is an urgent need to provide dynamic and integrated load-balancing solution to manage RFID network and serve upper layer applications. In our proposed model, the Edge server load-balancing module is responsible for monitoring the load on the Edge server and dynamic adjusting connections between Edge servers and RFID reader devices.

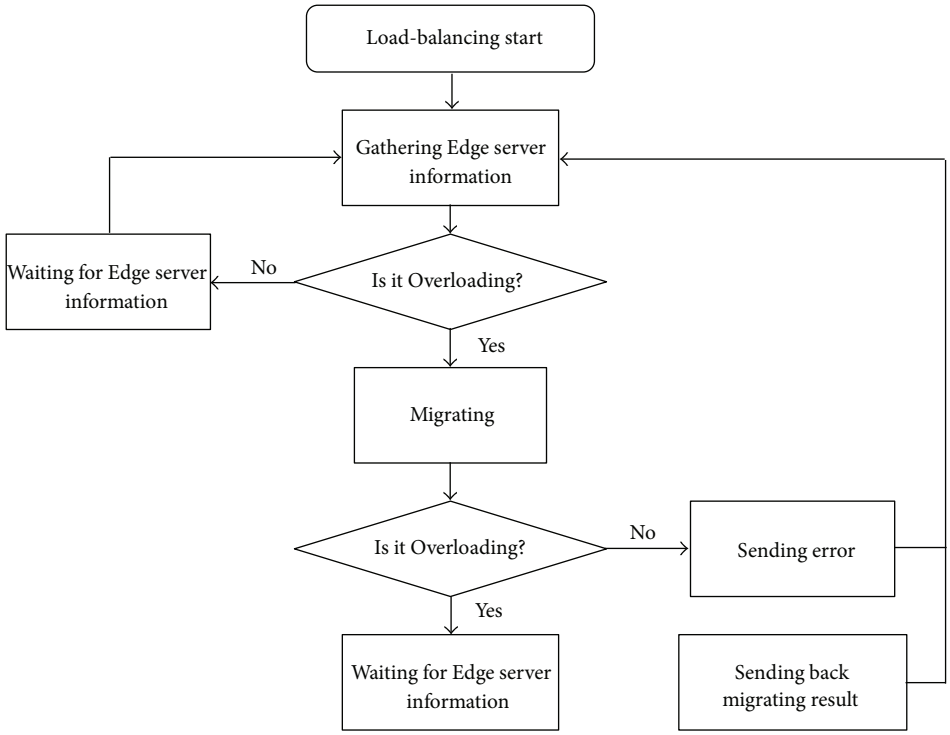


Figure 9: Edge server load balance mechanism.

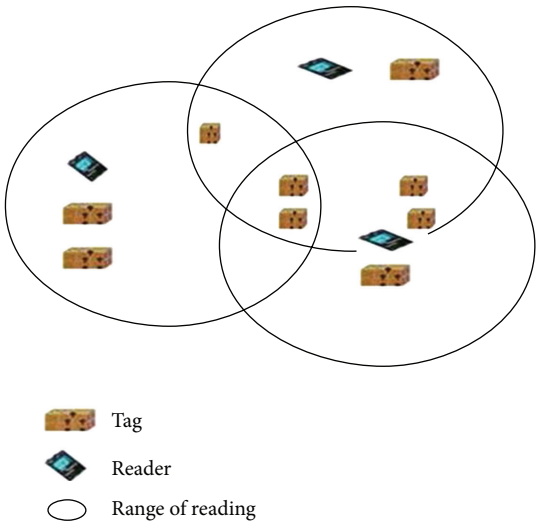


Figure 10: Data redundancy in RFID networks.

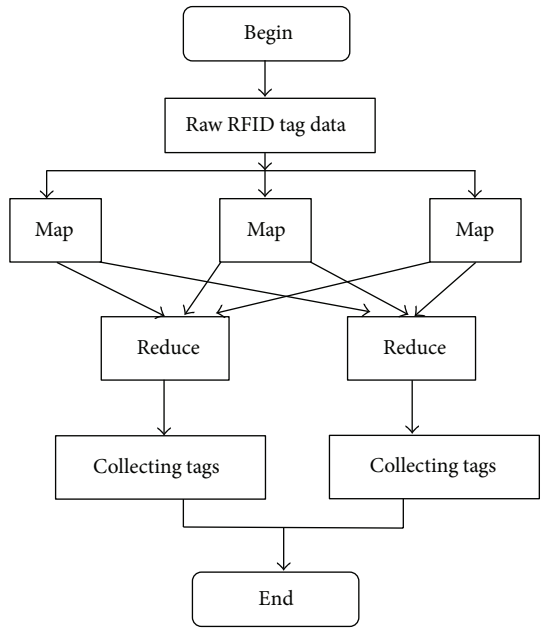


Figure 11: Proposed MapReduce flowchart.

Reference [28] defines an RFID network middleware system, reader collection, middleware collection, reader load, and middleware load as shown in Figure 12.

(a) $M = \{m_1, m_2, \ldots, m_n\}$, where M is a middleware collection and m_n represents middleware n; $CR_k = \{r_1^k, r_2^k, \ldots, r_p^k\}$ is a reader collection connected to the middleware k; one reader is connected to only one middleware during a period of time.

(b) $WL_R[r]$: the amount of tags handled by the reader, representing the load of a reader.

(c) $WL_M[m_i]$: the amount of tags handled by middleware m_i.

(d) $WL_M^U[m_i]$ and $WL_M^L[m_i]$ and the upper and lower load limits of middleware m_i, respectively.

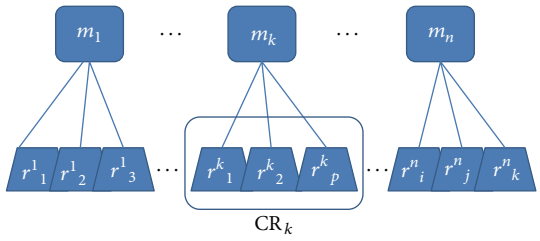


Figure 12: Definition of RFID middleware load [28].

Load-balancing strategy in [28] is to always choose the most overloaded middleware to migrate. During a certain period, relocation always migrates one or more readers on the most loaded middleware to least loaded middleware. This strategy is suitable for the situation that all middleware have the same configuration (homogenous case). This does not work for the case that middleware is not configured as the same. The configuration of middleware depends on CPU, memory, network bandwidth, and so forth, of their hosts. Therefore, we design a new strategy to consider the case that all middleware may not have the same configuration (heterogeneous case), with focus on dynamic and integrated load balancing.

(1) Average utilization of CPU and memory of each middleware:

$$\begin{aligned}
\text{CPU_}m_i &= \frac{WL_M[m_i]}{WL_M^U[m_i]} \times \text{CPU_}m_i^U + \text{CPU_}b,\\
\text{Mem_}m_i &= \frac{WL_M[m_i]}{WL_M^U[m_i]} \times \text{Mem_}m_i^U + \text{Mem_}b,\\
\text{AVG_}c &= \frac{\sum(\text{CPU_}m_i \times \text{SpeC_}m_i)}{\sum \text{SpeC_}m_i},\\
\text{AVG_}m &= \frac{\sum(\text{Mem_}m_i \times \text{SpeM_}m_i)}{\sum \text{SpeM_}m_i}.
\end{aligned} \tag{1}$$

(2) Integrated load imbalance level of middleware m_i is defined as

$$\begin{aligned}
&L_m[m_i]\\
&= \frac{1}{2}\left((\text{CPU_}m_i - \text{AVG_}c)^2 + (\text{Mem_}m_i - \text{AVG_}m)^2\right).
\end{aligned} \tag{2}$$

(3) Average imbalance level of host servers equals to the sum of imbalance levels of all servers divided by the number of hosting servers:

$$L\_M = \frac{1}{n}\sum L\_m[m_i]. \tag{3}$$

In (1)–(3), $\text{CPU\_}m_i$ is CPU utilization of host server m_i, $\text{Mem\_}m_i$ is memory utilization of host server m_i, AVG_c is average CPU utilization of all host servers, AVG_m is memory utilization of all host servers, $\text{SpeC\_}m_i$ is CPU configuration of host server m_i, for example, the number of CPU cores and CPU frequencies, and $\text{SpeM\_}m_i$ is memory configuration of host server m_i. $\text{CPU\_}m_i^U$ is CPU utilization of host server m_i with full load, and $\text{Mem\_}m_i^U$ is memory utilization of host server m_i with full load. CPU_b is CPU utilization of the middleware itself, and Mem_b is memory utilization of the middleware itself.

Integrated load balance considers factors including CPU utilization, memory utilization, and network bandwidth utilization, which can be expanded.

Allocation Policy. When there are new readers to add, it always chooses the middleware with lowest integrated load L_M to connect.

Reallocation (Migration) Policy. This mainly considers overloaded circumstance and chooses an overloaded middleware for migration. It needs to quantify how many readers to migration, at the same time to reduce the number of migrations to avoid system oscillation. Therefore, always migrate readers in the middleware with highest integrated load to the middleware with lowest integrated load until all load-balance thresholds (metrics) are met. In the migration, we take full advantages of live virtualization technology in Cloud computing. Migration of a virtual machine is simply moving the VM running on a physical machine (let us call it host Edge server) to another physical machine (let us call it target node). The key of live migration is that it does not disrupt any active network connections even after the VM is moved to the target node. Virtualization has other advantages such as running multiple applications on one Edge server, providing security and isolation to different applications, rapidly adjust resource allocation, and be transparent to applications without downtime. Figure 13 shows live migration of middleware by virtual machines.

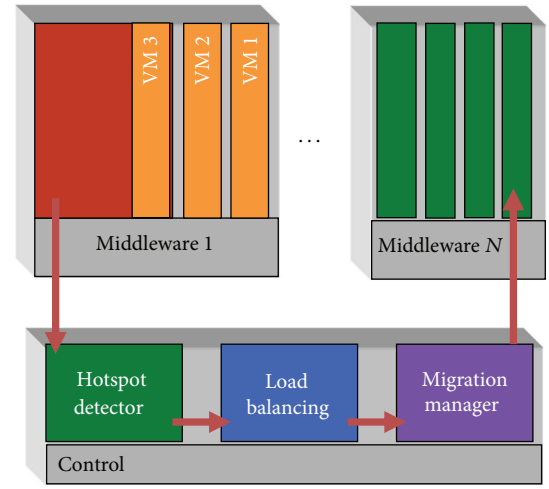


Figure 13: Live migration of middleware by virtual machines.

Through extensive numerical examples, we find that this dynamic and integrated load-balancing mechanism achieves

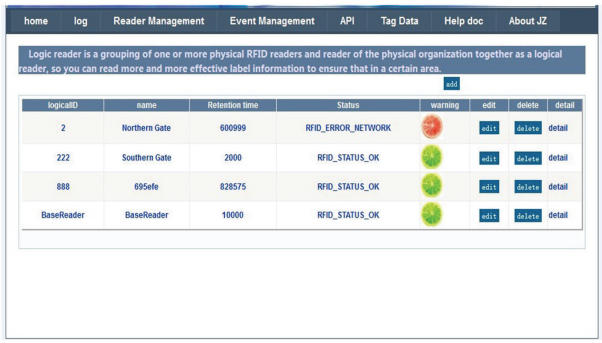


Figure 14: Logical view of RFID readers.

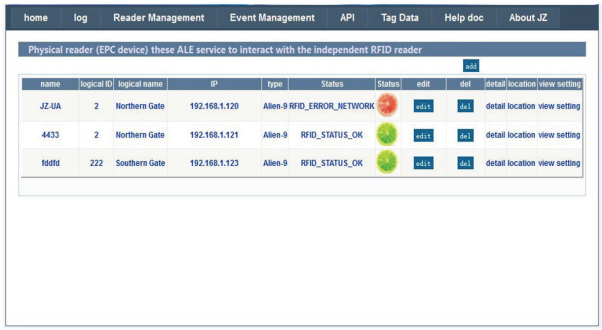


Figure 15: Physical RFID readers.

Figure 16: Tag data information.

Figure 17: ALE management.

lower average and a total imbalance level than traditional load-balancing strategies such as Round-robin and the one introduced in [1, 28].

3.6. Web Service APIs. In order to provide convenient management and service, Web service APIs are necessary. It has an interface described in a machine-processing format (e.g., WSDL format). Other systems can interact with the Web service APIs in a manner prescribed by description using messages, conveyed using HTTP with an XML serialization.

4. Performance Evaluation

4.1. General Information and Web APIs. Figures 14 and 15 show the logical view and physical view of readers, respectively; Figure 16 presents tag data information management, Figure 17 for ALE management, and Figure 18 for Web service APIs.

4.2. Load Balance. For the performance evaluation, we have the following configuration: Edge server A has one reader with process rate of 1500 tags/second; Edge server B has two readers, both with process rate 2000 tags/second; Edge server C has three readers with process rate 5000 tags/second; Edge server D has one reader with process rate 6000 tags/second. Table 1 also provides the configuration of host Edge servers including CPU in Ghz, memory (MEM) in GB; the upper bound (UB) of utilization of CPU (UB) and MEM (UB); and $WL_M^U[m_i]$ for each host Edge servers. Table 2 is the CPU and MEM utilization information of four host Edge servers before load balancing. Table 3 provides CPU and MEM utilization results comparison between ARRMS and the method in [28]. It can be observed that ARRMS has better integrated load balance for both CPU and memory in host Edge servers since it considers CPU and memory integrated.

4.3. Redundancy Elimination and Others. Table 4 shows the data format of an RFID tag: Device ID (2 bits) represents the identification of the device, EPC code (28 bits) is for tag content, and Timestamp (14 bits) records date and time information of a tag.

Taking a tag "003000C2001602200001457FFC000420110627172133" as an example:

(i) bit 0-1-00 is the device number of a tag, which has length 2;

(ii) bit 2-29-3000C2001602200001457FFC0004 are tag's EPC code, whose length is 28 bits;

(iii) bit 30-43-20110627172133 is the information of timestamp of a tag, 20110627 is the date, 172133 is the time, and the length of the time is 14 bits.

First, we define data redundancy based on the following RFID data model [25]: when a reader reads a tag, the following data can be read: EPC (EPC code), reader (reader ID), and timestamp (timestamp). When the reader reads two

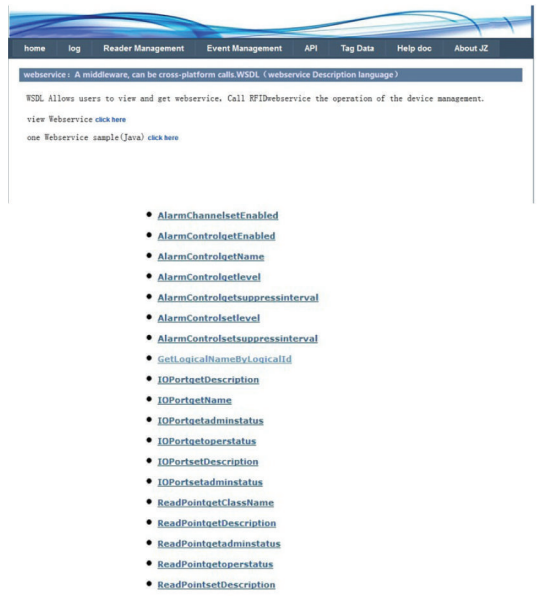


Figure 18: Some Web service APIs.

Table 1: The configuration of Edge servers.

Edge server	CPU (GHz)	MEM (GB)	CPU (UB)	MEM (UB)	$WL_M^U[m_i]$
A	1.5	1	0.7	0.7	5000
B	2	1.5	0.7	0.7	10000
C	2.5	2	0.7	0.7	20000
D	3	2.5	0.7	0.7	30000

Table 2: Utilization information before load balancing.

Edge server	CPU utilization (%)	MEM utilization (%)
A	34	37
B	42	46
C	70	78
D	26	28

labels (Label A and Label B), if all of the following three conditions are satisfied, it can be said that these two labels are repeated data (tags), that is redundant data [4]:

Label A: EPCA, reader A, and timestamp A;

Label B: EPCB, reader B, and timestamp B.

(1) EPC codes are the same, that is EPCA = EPCB.

(2) Reader A and Reader B are the same.

(3) The difference between Timestamp A and Timestamp B is less than constant T, which may be 10 milliseconds, for example. Figure 19 provides an example using MapReduce.

The process of parallel MapReduce redundancy data elimination can be summarized as follows (see [27] for more details).

(1) In Map phase, the data structure Tag consists of three variables: Reader ID, EPC, and Timestamp. According to the definition of tag redundancy, if two tags are redundant, their Reader IDs and EPC codes must be the same, and if one of them is different, the two tags are not redundant tags. Input RFID tags are divided into different blocks and assigned to hosts of Hadoop clusters. Reader ID and EPC codes are Keys and timestamps are outputs as Values to form (Key, Value) pairs in MapReduce model. In this case, that is, Key is (Reader ID + EPC), while Value is Timestamp.

(2) After simple sorting based on the Key, Map (mapping) outputs of each host are transferred to Reduce.

(3) Reduce stage: based on outputs of Map stage, it compares values of each group of Key Value (calling time comparison function), and if they are less than constants T (predefined), they are redundant and should be removed, then Reduce removes the redundancies and obtains an output and repeats doing this until all inputs are proceeded. Notice that there may be multiple workers (servers) that work paralleled in Map and Reduce stages.

(4) Working nodes (hosts) in MapReduce cluster summarize *Key*, *Value* pairs of other hosts processed by Reduce and output them according to the original tag format.

Table 3: Utilization information after applying our load-balancing mechanism and method in [28].

Edge server	CPU (%) [28]	MEM (%) [28]	CPU (%) ARRMS	MEM (%) ARRMS
A	34.0	37.0	34.0	37.0
B	82.0	91.0	42.0	46.0
C	50.0	55.0	50.0	55.0
D	26.0	28.0	39.0	43.0
L_M	0.12		0.01	

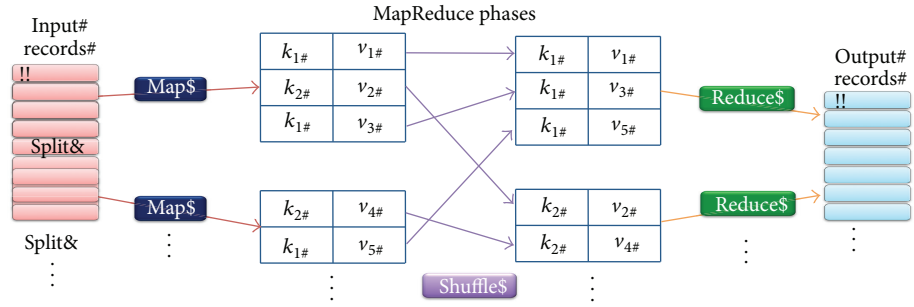


Figure 19: Redundancy elimination example using MapReduce.

Table 4: Data format of an RFID tag.

Field	Device ID	EPC code	Timestamp
Bits	2	28	14

As for evaluation, set the total number of RFID tags is S, the redundancy ratio is P_d and redundant tags are randomly generated following EPC data format (32 bits currently), as shown in Table 4 where total number of tags varying from 30 to 45 million and Cluster size means the number of host servers used in Hadoop MapReduce. From Table 5, we can see that the larger the number of tags and the MapReduce cluster size is, the bigger the difference of time spent on redundancy elimination is. As the number of tags increases (above 50000), MapReduce begins to show the advantages of scalability. The proposed method can be applied to large scale RFID applications such as supermarkets and other areas conveniently. The reason that Hadoop cluster can remove 100% redundant data for large-scale of data up to 100 million lies in that it is a centralized redundancy elimination mechanism and it can take all redundancy information into consideration.

Figure 20 also shows tag lost ratio comparison from simulation results, where connection pool is obtained using the method in [30] and Agent is obtained, applying by method in [29]. In the simulation, the total number of readers is fixed as 30 but with varying the total number of middleware: 2, 4, 6, 8, and 10. The tag information generation rate is 50 tags/sec, and the time for a middleware to process a tag information is 2 microseconds, and the total test length of each run is 15 minutes, and our results are the average of six runs. Figure 21 also provides average process time comparison for the same total amount of tags. Similar results are obtained in other cases, because of page limit, those results are not provided here.

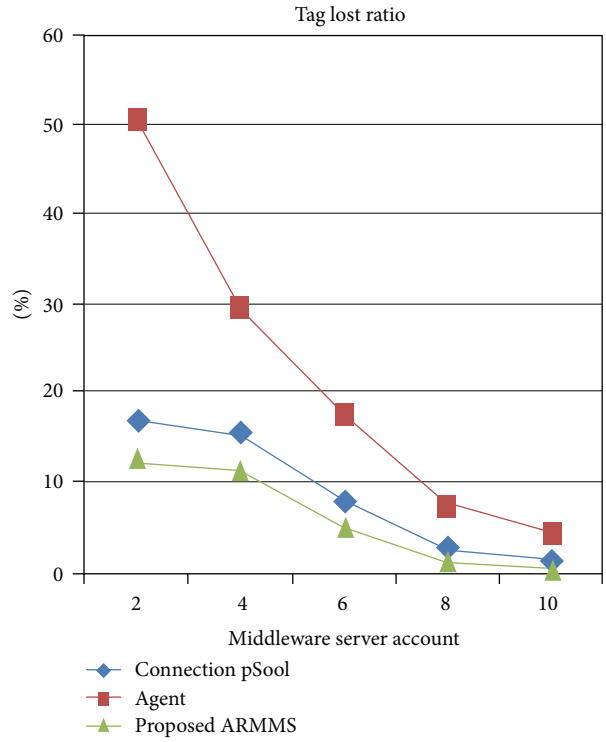


Figure 20: Tag lost ratio comparison.

5. Conclusions and Future Work

In this paper, a distributed and messaging RFID middleware design is proposed. ARMMS provides equipment management, RFID tag information management, application level event (ALE) management, and Web service APIs, and so forth, functions. For the equipment management, ARMMS

Table 5: Times spent for Hadoop cluster to eliminate redundancy.

Number of tags	Cluster size						
	1 server	2 servers	3 servers	4 servers	10 servers	15 servers	20 servers
30	4 sec	7 sec	7 sec	7 sec	7 sec	7 sec	7 sec
50000	5 sec	8 sec	7 sec	7 sec	4 sec	2 sec	1 sec
500000	16 sec	18 sec	14 sec	12 sec	6 sec	3 sec	2 sec
1.2 million	33 sec	24 sec	22 sec	19 sec	10 sec	6 sec	3 sec
45 million	52 min	48 min	42 min	41 min	21 min	11 min	5 min

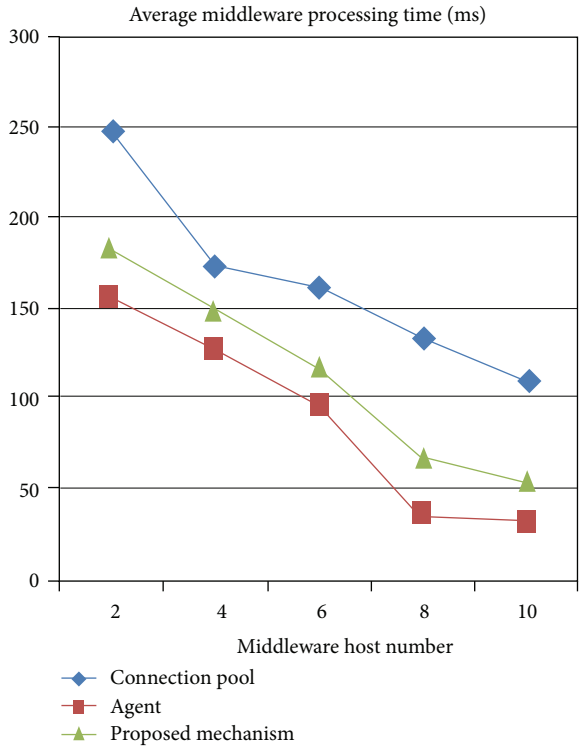


Figure 21: Average process time comparison.

provides real-time monitoring and load balancing of multi-site RFID devices also logical grouping of RFID readers. For tag information management, it provides filtering and statistical reports. ARMMS also provides Web service APIs based on EPCglobal standards. We are conducting more experimental tests and comparing different load-balancing strategies with proposed one. There are still several research directions awaiting further investigation.

(1) Considering distributed redundancy data elimination: when facing large-scale distributed applications, distributed redundancy data elimination other than parallel method should be investigated further, and elimination efficiency should be quantitatively evaluated.

(2) Considering real-time data allocation and evaluating further live migration costs and load balancing: real-time data allocation among different middleware (host Edge servers) causes a new challenge for load balance and should be considered further. Also, the live migration costs in time and other respects should be included to provide a complete view for load balance so that decision makers can have comprehensive information.

(3) Providing more comparative results against existing methods: currently, there is still difficulty to repeat some of the existing methods such as in [1], where experimental details are not provided. We need repeating their algorithms and configuration so that more comparative scienarios and results can be obtained.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (NSFC) (Grants nos. 61150110486, 61034005 and 61272528) and by China Postdoc Funding (2011-2012).

References

[1] Y. Ma, H. Chao, J. Chen, and C. Wu, "Load-balancing mechanism for the RFID middleware applications over grid networking," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 811–820, 2011.

[2] I. Abad, C. Cerrada, J. A. Cerrada, R. Heradio, and E. Valero, "Managing RFID sensors networks with a general purpose RFID middleware," *Sensors*, vol. 12, no. 6, pp. 7719–7737, 2012.

[3] C. Floerkemeier and M. Lampe, "RFID middleware design: addressing application requirements and RFID constraints," in *Proceedings of the Smart Objects Conference (SOC '05)*, pp. 219–224, Grenoble, France, October 2005.

[4] EPCglobal Inc, http://www.epcglobalinc.org/.

[5] X. Huang, S. Le, and D. Sharma, "A taxonomy for RFID systems," in *Proceedings of the 1st International Conference on Signal Processing and Communication System*, pp. 1–8, Gold Coast, Australia, December 2007.

[6] M. A. E. Khaddar, M. Boulmalf, H. Harroud, and M. Elkoutbi, "RFID middleware design and architecture, book chapter, design and deploying RFID applications".

[7] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.

[8] R. Weinstein, "RFID: a technical overview and its application to the enterprise," *IT Professional*, vol. 7, no. 3, pp. 27–33, 2005.

[9] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*, John Wiley & Sons, New York, NY, USA, 2000.

[10] B. Nath, F. Reynolds, and R. Want, "RFID technology and applications," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 22–24, 2006.

[11] "ALE_1_1_1-standard-core-20090313[EB/OL]," http://wwwepcglobalinc.org, 2009.

[12] "ALE_1_1_1-standard-XML_and_SOAP_bindings-20090313[EB/OL]," http://www.epcglobalinc.org, 2009.

[13] Auto-ID Labs, http://www.autoidlabs.org/S .

[14] "Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1. 0. 9[EB/OL]," http://www.epcglobalinc.org/.

[15] "EPCglobal-ReaderManagementrm_1_0_1-standard-20070531 [EB/OL]," http://www.epcglobalinc.org/, 2007.

[16] B. Feng, J. T. Li, P. Zhang, and J. B. Guo, "Study of RFID middleware for distributed large-scale systems," in *Proceedings of the Information and Communication Technologies (ICTTA '06)*, vol. 2, pp. 2754–2759, 2006.

[17] M. Cezon, G. Vaudaux-Ruth, L. Laurens, and J. Soldatos, "Review of state-of-the-art middleware," ASPIRE Project Public Deliverable D2. 1., 2008.

[18] F. Lin and B. Chen, "The design of a lightweight RFID middleware," *International Journal of Engineering Business Management*, vol. 1, no. 2, pp. 25–30, 2009.

[19] B. S. Prabhu, X. Su, H. Ramamurthy, C. Chu, and R. Gadh, "WinRFID: a middleware for the enablement of radio frequency identification (RFID) based applications," in *Proceedings of the Wireless Internet for the Mobile Enterprise Consortium (WINMEC '05)*, Los Angeles, Calif, USA, December 2005.

[20] P. Tran, P. Greenfield, and I. Gorton, "Behavior and performance of message-oriented middleware systems," in *Proceedings of the 22nd International Conference on Distributed Computing Systems*, 2002.

[21] Y. Bai, F. Wang, and P. Liu, "Efficiently filtering RFID data streams," in *Proceedings of the 1st International VLDB Workshop on Clean Databases (CleanDB '06)*, September 2006.

[22] C. Yingwen, V. L. Hong, X. Ming, C. Jiannong, K. C. C. Chan, and A. T. S. Chan, "In-network data processing for wireless sensor networks," in *Proceedings of the 7th International Conference on Mobile Data-Management (MDM '06)*, Nara, Japan, May 2006.

[23] W. Choi and M. Park, "In-network phased filtering mechanism for a large-scale RFID inventory application," in *Proceedings of the 4th International Conference on Information Technology and Applications (ICITA '07)*, pp. 401–405, Harbin, China, January 2007.

[24] S. R. Jeffery, M. Garofalakis, and M. J. Franklin, "Adaptive cleaning for RFID data streams," in *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06)*, September 2006.

[25] I. Kadayif and M. Kandemir, "Tuning in-sensor data filtering to reduce energy consumption in wireless sensor networks," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE '04)*, pp. 1530–1539, Paris, France, February 2004.

[26] A. K. Bashir, S. Lim, C. S. Hussain, and M. Park, "Energy efficient in-network RFID data filtering scheme in wireless sensor networks," *Sensors*, vol. 11, no. 7, pp. 7004–7021, 2011.

[27] W. H. Tian, Y. P. Yang, K. She, X. Dong, and H. Y. Wang, "A parallel method to redundancy data elimination in RFID network," in *Proceedings of the International Conference on Instrumentation, Measurement, Circuits and Systems (ICIMCS '11)*, pp. 399–402, Hongkong, China, December 2011.

[28] H. S. Chae and J. Park, "An approach to adaptive load balancing for RFID middlewares," *International Journal of Mathematical and Computer Sciences*, vol. 2, no. 2, 2006.

[29] F. C. Jian and S. C. Heung, "Agent-based design of load balancing system for RFID middlewares," in *Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS '07)*, pp. 21–28, March 2007.

[30] S. Park, J. Song, C. Kim, and J. Kim, "Load balancing method using connection pool in RFID middleware," in *Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management, and Applications (SERA '07)*, pp. 132–137, August 2007.

[31] J. G. Park, H. S. Chae, and E. S. So, "A dynamic load balancing approach based on the standard RFID middleware architecture," in *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE '07)*, pp. 337–340, October 2007.

[32] W. H. Tian, K. She, Y. P. Yang, and X. Dong, "An approach to dynamic and integrated load-balancing of distributed and messaging RFID middleware," in *Proceedings of the International Conference on Instrumentation, Measurement, Circuits and Systems (ICIMCS '11)*, pp. 403–406, Hongkong, China, December 2011.

[33] J. Dean and S. Ghemawat, "MapReduce: simplied data processing on large clusters," Google Paper, 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 120527, 14 pages
http://dx.doi.org/10.1155/2013/120527

Research Article

Optimal QoM in Multichannel Wireless Networks Based on MQICA

Na Xia,[1,2] Lina Xu,[1] and Chengchun Ni[1]

[1] *School of Computer and Information, Hefei University of Technology, Hefei 230009, China*
[2] *Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education of China, Hefei 230009, China*

Correspondence should be addressed to Na Xia; xiananawo@hfut.edu.cn

Received 28 December 2012; Accepted 13 May 2013

Academic Editor: Xinrong Li

In wireless networks, wireless sniffers are distributed in a region to monitor the activities of users. It can be applied for fault diagnosis, resource management, and critical path analysis. Due to hardware limitations, wireless sniffers typically can only collect information on one channel at a time. Therefore, it is a key topic to optimize the channel selection for sniffers to maximize the information collected, so as to maximize the Quality of Monitoring (QoM) for wireless networks. In this paper, a Multiple-Quantum-Immune-Clone-Algorithm- (MQICA-) based solution was proposed to achieve the optimal channel allocation. The extensive simulations demonstrate that MQICA outperforms the related algorithms evidently with higher monitoring quality, lower computation complexity, and faster convergence. The practical experiment also shows the feasibility of this algorithm.

1. Introduction

With the growing application of wireless networks (e.g., WiFi, WiMax, Mesh, and WLAN), high quality management of wireless device and networks is becoming more and more important [1–3]. It has been a key point to monitor network status and performance accurately and in real time, so as to implement effective management.

Wireless monitoring is usually realized using Simple Network Management Protocol (SNMP) and base-station logs. Since they reveal detailed PHY (e.g., signal strength and spectrum density) and MAC behaviors (e.g., collision and retransmission), as well as timing information, they are essential for network diagnosis and management [4–9]. But wireless monitoring equipments are usually single-radio multichannel device [10–12]. That is to say, it has multioptional channels (In IEEE 802.11.b/g WLAN, there are 3 orthogonal channels, and in IEEE 802.11.a WLAN, there are 12 orthogonal channels). So, it is a key topic to allocate channels and other resources for these monitoring equipments to optimize the monitoring quality of entire network [13–17]. In the literature [16], it has turned out to be a NP-hard problem in user-center mode, and an effective solution for the problem will be with great significance to the performance improvement of all kinds of wireless application networks.

In this paper, we carry out the full investigation on the current wireless monitoring networks and establish a system monitoring model based on the undirected bipartite graph. Then, compared with existing algorithms, we propose an optimization solution "Multiple Quantum Immune Clone Algorithm (MQICA)" to solve the problem. Finally, the algorithm has been proved to be with good performance both in theory and experiments.

The rest of the paper is organized as follows. In Section 2, we provide a brief review of existing work on wireless monitoring. The problem formulation is presented in Section 3. The Multiple Quantum Immune Clone channel allocation algorithm (MQICA) is detailed in Section 4. Then we prove the validity of the proposed algorithm in Section 5 followed by extensive simulation experiments in Section 6. Finally, we conclude this paper with some future work in Section 7.

2. Related Work

In recent years, wireless monitoring networks have become a hot topic. The research mainly contains monitoring device, system design, fault diagnosis and so forth [4–9]. In 2004, "passive monitoring" utilizing multi-wireless sniffers was first introduced by Yeo et al. [4, 5]. He analyzed the advantages and challenges of wireless passive monitoring and preliminarily set up an application system, which fulfilled the network fault diagnosis based on time synchronization and data fusion of multisniffers. In 2005, Rodrig et al. [6] used sniffers to capture wireless communication data and analyze the performance characteristics of 802.11 WiFi network. In 2006, Cheng et al. [7] investigated a large-scale monitoring network composed of 150 sniffers and discussed the time synchronization method for distributed sniffers. In 2007, Yang and Guo et al. [8] studied the lifetime model of wireless monitoring networks and proposed to adjust the sensing and communication radius of sniffers in real time to maximize the lifetime of networks. In 2010, Liu and Cao [9] researched the relationship between the number of monitoring sniffers and false alarm rate and put forward an algorithm based on *poller-poller* structure, which can limit the false alarm rate and minimize sniffers.

It has become an important subject to optimize the channel selection of monitoring sniffers so as to improve the network monitoring quality. In 2009, Shin and Bagchi [13] researched the channel selection of sniffers in Wireless Mesh network to maximize the coverage of users. He described it as a maximum coverage problem based on *group budget constraints* [14, 15] and solved it using Greedy and Linear Programming (LP) algorithms, which achieved good performance. Based on the previous research, Chhetri et el. [16] formulated the problem of channel allocation of sniffers and proved it to be NP-hard to maximize the Quality of Monitoring (QoM) of wireless network under universal network model. Greedy and LP algorithms were employed to solve the problem. Greedy algorithm always seeks the solution with maximal current benefit during the process of resolution and misses the global optimal solution or approximate of it. Although LP algorithm can achieve better solution than others, its complexity is too high to meet the real-time optimization in dynamic wireless networks. In 2011, we applied *Gibbs Sampler* theory to address the problem and proposed a distributed channel selection algorithm for sniffers to maximize QoM of network [17]. This method utilizes the local information to select the channel with low energy but cannot achieve the global optima in most of the cases.

In [15–20], we can get an overview of much excellent work in multichannel selection of wireless network itself. In 2006, Wormsbecker and Williamson [18] studied the impact of channel selection technique on the communication performance of system and applied *soft channel reservation* technique to select channels, so as to reduce link layer data frame losses and provide higher TCP throughput. In 2007, Kanthi and Jain [19] proposed a channel selection algorithm for multiradio and multichannel mesh networks. It is based on *Spanner* conception and combined with network topology. The experiment results showed that it can improve data throughput in communication link layer. In 2009, You et al. [20] investigated the end-to-end data transmission and the optimal allocation of channel resource in wireless cellular networks and figured it out with *stochastic quasi-gradient* method. In 2010, Hou and Huang [21] researched the channel selection problem in Cognitive Ratio networks, described it as a binary integer nonlinear optimization problem, and proposed an algorithm based on *priority order* to maximize the total channel utilization for all secondary nodes. In 2011, an interface-clustered channel assignment (ICCA) scheme was presented by Du et al. [22]. It can eliminate the collision and interference to some extent, enhance the network throughput, and reduce the transmission delay.

From what has been discussed previously, there exist great shortcomings in solution of wireless monitoring network channel allocation problems. All of these studies most focused on the wireless network itself, rather than the wireless monitoring sniffers. Existing algorithms will have high algorithm complexity, slow convergence speed, and, in most cases, it is difficult to get global optimal solution in the cases of large-scale networks or having more optional channels. Therefore, in this paper, a so-called Multiple-QICA (MQICA) algorithm, taking full advantage of the parallel characteristics of Quantum Computing (QC), is proposed. Compared with traditional Quantum Immune Clone Algorithm (QICA), MQICA possesses lots of characteristics inherited from both immune and evolution algorithms. Meanwhile, allele and Gaussian mutations are introduced in MQICA to further improve the performance of the algorithm. Extensive simulations and practical experiments demonstrate that the proposed algorithm outperforms other algorithms not only in quality of solution, but also in time efficiency.

3. Problem Description

3.1. Network Model. Consider a wireless network of m monitoring sniffers, n users, and k optional channels. $S = \{s_1, s_2, \ldots, s_m\}$ is the set of sniffers, $U = \{u_1, u_2, \ldots, u_n\}$ is the set of users, and $C = \{c_1, c_2, \ldots, c_k\}$ is the set of channels. In homogeneous networks, sniffers have the same transmission characteristics. They have the ability to read frame information and can analyze the information from users or other sniffers. But at any point in time, a sniffer can only observe transmissions over a single channel. Let p_{u_j} denote the transmission probability of a user u_j ($j = 1, 2, \ldots, n$) that works on channel $c(u_j) \in C$. These users can be a wireless router, access point or mobile phone user, and so forth. If a user sends data through a channel at time t, it will be called active user in time t.

In wireless networks, the relationship between sniffers and users can be described by an undirected bipartite graph $G = (S, U, E)$ shown in Figure 1. If u_j is in the monitoring area of s_i, there will be a connection between them, indicated by $e = (s_i, u_j)$. When s_i and u_j work on the same channel, s_i can capture the data from u_j, and then we say that u_j is covered by s_i. E represents the set of all connecting edges. If a user is outside all sniffers' monitoring area, it is excluded from G.

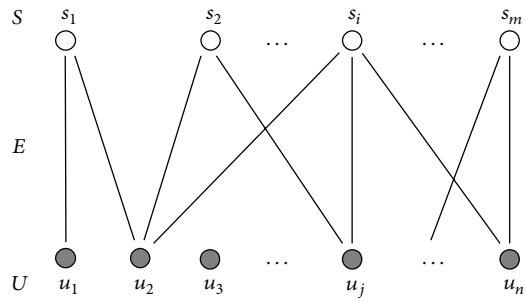


Figure 1: undirected bipartite graph G.

The vertex v in G is sniffer or user, namely, $v \in S \cup U$. $N(v)$ denotes the neighbors of vertex v. If the vertex is a user u_j, $N(u_j)$ means its neighbor sniffers; if the vertex is a sniffer s_i, $N(s_i)$ is the set of neighbor users of s_i. If a sniffer is inside the communication range of another sniffer, they are called adjacent sniffers. $V(s_i)$ denotes the set of adjacent sniffers of s_i, and B_{s_i} is the set of subscript of sniffers in $V(s_i)$. In this paper, we assume that the communication radius of sniffer is twice as its monitoring radius.

3.2. Problem Formulation. $\mathbf{a} : S \rightarrow C$ represents a channel selection scheme for wireless monitoring networks, and O is the set of all possible schemes. $\mathbf{a}$ can be expressed in the form of vector as follows: $\mathbf{a} = (a(s_1), a(s_2), \ldots, a(s_m))$, where $a(s_i) \in C$ is the channel selected by s_i. When s_i selects the channel $a(s_i)$, it can communicate with the neighbor users, who also work on channel $a(s_i)$. Given a channel selection scheme $\mathbf{a}$, then $S = \cup_{q=1}^{k} S_{c_q}$, $U = \cup_{q=1}^{k} U_{c_q}$, where S_{c_q} denotes the set of sniffers assigned to channel c_q, and U_{c_q} denotes the set of users working on channel c_q. Now it is able to show the relationship between all the sniffers and users working on channel c_q in the form of undirected bipartite graph $G_{c_q} = (S_{c_q}, U_{c_q}, E_{c_q})$.

Definition 1. Monitoring quality of node (MQN): when wireless monitoring network works on channel $\mathbf{a} \in$ O, the monitoring quality of node s_i can be defined as follows:

$$Q_{s_i}(\mathbf{a}) = \sum_{u \in N(s_i)} p_u \cdot \frac{\mathbf{1}(c(u) = a(s_i))}{1 + \sum_{t \in B_{s_i}} \mathbf{1}(c(u) = a(s_t), s_t \in N(u))}, \tag{1}$$

where $\mathbf{1}(\cdot)$ is an indicator function. It equals 1 when the condition is true and 0 otherwise. It is clear that the more neighbor users work on the same channel as s_i, the higher transmission probability these users have, and meanwhile, the less other sniffers cover these users, the higher monitoring quality s_i has. MQN reflects the number of active users available to s_i under the channel selection scheme $\mathbf{a}$. Active users are in the state of sending data.

Given a channel selection scheme $\mathbf{a}$, the Quality of Monitoring (QoM) of wireless network can be defined as follows:

$$Q(\mathbf{a}) = \sum_{s_i \in S} Q_{s_i}(\mathbf{a}). \tag{2}$$

So, the higher QoM is, the more active users can be monitored in the network and the higher quality of service the wireless monitoring network provides.

The problem of maximizing of QoM (MQM) can be described as follows: finding an optimal channel allocation scheme for sniffers to collect the largest amount of information transmitted by users, that is, to maximize the QoM of the network.

The channel allocation scheme will be changed according to probability during different time slot. So the maximal information collected by monitoring network in a certain period can be expressed as

$$\begin{aligned} \max \quad & \sum_{\mathbf{a} \in O} Q(\mathbf{a}) \times \pi(\mathbf{a}), \\ \text{s.t.} \quad & \pi(\mathbf{a}) \in [0, 1] \\ & \sum_{\mathbf{a} \in O} \pi(\mathbf{a}) = 1, \end{aligned} \tag{3}$$

where $\pi(\mathbf{a})$ is the probability for wireless monitoring network to work on the channel allocation scheme $\mathbf{a}$.

From (3), the optimal channel allocation scheme will be got as follows:

$$\mathbf{a}^* = \arg\max Q(\mathbf{a}). \tag{4}$$

For this complicated combination optimization problem, an effective heuristic algorithm is needed. In 2005 Jiao and Li proposed a brand new Quantum-Inspired Immune Clone Algorithm (QICA) [23]. QICA constructs antibodies in view of the superposition characteristics of quantum coding and enlarges the original population via clone operation, thus expanding the searching space and improving the performance of the algorithm when doing local search. It is very suitable for this complicated combination optimization problem because of the attributes of parallelism and provable rapid convergence. But the results in QICA are expressed in a binary form [24], which are more appropriate for solving problems in a binary encoding. Thus we need to extend it to k-resolution coding before applying the algorithm to this MQM problem in wireless monitoring network. Then, Multiple-QICA channel selection algorithm is proposed and described in detail as follows.

4. Multiple Quantum Immune Clone Channel Allocation Algorithm (MQICA)

4.1. Fundamental Definitions. To accurately describe the evolutionary process of the MQICA algorithm, the following fundamental definitions are proposed.

Definition 2 (Channel Quantum Antibody (CQA)). We define the *Channel Quantum Antibody* as the following triploid chromosome:

$$\mathrm{CQA} \overset{\text{def}}{=} \begin{bmatrix} x_0 \cdots x_i \cdots x_{m-1} \\ \alpha_0 \cdots \alpha_i \cdots \alpha_{m-1} \\ \beta_0 \cdots \beta_i \cdots \beta_{m-1} \end{bmatrix}, \tag{5}$$

where m is called the length of the chromosome, that is, the number of the monitoring sensors. $x_i \in [0, 1)$ represents channel selection scheme of the ith monitoring sensor. α_i and β_i should meet the normalization condition: $|\alpha_i|^2 + |\beta_i|^2 = 1$, where $|\alpha_i|^2$ and $|\beta_i|^2$ indicate, respectively, the nonoptimal and optimal probability of the channel selection scheme of the ith monitoring sensor. $\begin{bmatrix} x_i & \alpha_i & \beta_i \end{bmatrix}^T$ is named as an allele of the CQA.

Definition 3 (mapping between antibody to channel). Note that in the CQA,

$$X = \left[x_0, x_1, \ldots, x_i, \ldots, x_{m-1}\right] \in R^m, \tag{6}$$

where $x_i \in [0, 1)$ is a continuous real number. If it is discrete, X can be mapped into integer space that

$$\mathbf{C} = \left[c_{s_0}, c_{s_1}, \ldots, c_{s_i}, \ldots, c_{s_{m-1}}\right] \in Z^m, \tag{7}$$

where $c_{s_i} \in \{0, 1, \ldots, k-1\}$ and indicates the monitoring sensor s_i to select channel c_{s_i}. k is the total number of selectable channels in the network. The process described previously is called mapping of the CQA to channel selection scheme, briefly as antibody to channel. The *mapping relationship* is defined as follows:

$$c_{s_i} \overset{\text{mapping}}{=} \left\lfloor k x_i \right\rfloor, \quad i = 0, 1, \ldots, m-1. \tag{8}$$

Definition 4 (channel affinity). *Channel affinity* refers to the affinity degree between the CQA and the channel antigen, which is the approximate level between feasible solution and optimal solution. With the affinity value increased, the feasible solution will be much closer to the optimal one. On the contrary, the feasible solution will gradually deviate from the optimal one. *Channel affinity* is the foundation of immune selection operation.

Definition 5 (evolutionary entropy of the CQA). To measure the extent of the evolution, we introduce *evolutionary entropy* to the CQA:

$$\begin{aligned} H(X) &= H\left(x_0, x_1, \ldots x_{m-1}\right) \\ &= H\left(x_0\right) + H\left(x_1 \mid x_0\right) \\ &\quad + \cdots + H\left(x_{m-1} \mid x_0, x_1, \ldots, x_{m-2}\right) \\ &= \sum_{i=0}^{m-1} \left(|\alpha_i|^2 \log \frac{1}{|\alpha_i|^2} + |\beta_i|^2 \log \frac{1}{|\beta_i|^2} \right) \\ &= -\sum_{i=0}^{m-1} \left(|\alpha_i|^2 \log |\alpha_i|^2 + |\beta_i|^2 \log |\beta_i|^2 \right) \\ &= -2 \sum_{i=0}^{m-1} \left(|\alpha_i|^2 \log |\alpha_i| + |\beta_i|^2 \log |\beta_i| \right). \end{aligned} \tag{9}$$

As the evolution process continues, $\alpha_i \to 0$, $\beta_i \to 0$, thus $H(X) \to 0$. So the *evolutionary entropy* can be used to describe the extent of the evolution. When the algorithm finally comes to a convergent result, the value of *evolutionary entropy* is indefinitely close to zero.

4.2. Process Design. The population is denoted as $A = \{r_1, r_2, \ldots, r_N\}$, where N indicates the scale of the population and r_i represents a CQA in it. An evolution process of the algorithm in this paper consists of three basic operations, including clone, immune genetic variation, and immune selection. Clone operation (T_c) clones each antibody and the clone scale is decided by the *channel affinity* value of the antibody. Immune genetic variation (T_m) will increase the diversity of population information. The immune selection operation (T_I) chooses from all antibodies generated by the former two operations according to their *channel affinity* and get the optimal CQA. Then compare them with the original N elite antibody r''_{i0} [25], $i = 1, 2, \ldots, N$ in immune memory set S_m [26]. Meanwhile, it forms the new population of next generation. Thus an evolution process can be described as

$$\begin{aligned} &A(t) \xrightarrow{\text{Clone Operation } (T_c)} A'(t) \xrightarrow{\text{Immune Genetic Variation } (T_m)} \\ &A''(t) \xrightarrow{\text{Immune Selection } (T_I)} A(t+1). \end{aligned} \tag{10}$$

After this operation, we need to do full interference cross to the new operation $A(t+1)$ and continue returning to the next, in case that the evolution still did not meet the termination conditions.

4.2.1. Cloning Operation. A self-adaptive clone operation is proposed in [23]:

$$T_c\left(A(t)\right) = \left\{T_c\left(r_1\right), T_c\left(r_2\right), \ldots, T_c\left(r_N\right)\right\}, \tag{11}$$

where $T_c(r_i) = E_i \times r_i$, $i = 1, 2, \ldots, N$ and E_i is a unit row vector with q_i columns, while q_i indicates the clone scale of the CQA and is decided by the equation as follows:

$$q_i = \left\lceil N_c \times \frac{Q\left(\mathbf{C}_i\right)}{\sum_{j=1}^{N} Q\left(\mathbf{C}_j\right)} \right\rceil, \tag{12}$$

where $N_c > N$. It can be concluded from (12) that if the channel affinity of a specific CQA is greater than that of the others in the population, then corresponding clone scale will be larger. Thus this clone strategy guarantees that the more excellent an antibody is, the more resource it will get, and this will obviously drive the algorithm to evolve towards the optimal solution much more quickly. Once the clone operation is completed, the population $A(t)$ is expanded to have the following form:

$$A'(t) = \left\{A(t), A'_1(t), \ldots, A'_N(t)\right\}, \tag{13}$$

where,

$$\begin{aligned} A'_i(t) &= \left\{r_{i1}(t), r_{i2}(t), \ldots, r_{iq_i-1}(t)\right\}, \\ r_{ij}(t) &= r_i(t), \quad j = 1, 2, \ldots, q_{i-1}. \end{aligned} \tag{14}$$

4.2.2. Immune Genetic Variation. MQICA algorithm implements a single-gene mutation on every triploid chromosome during the evolution. Compared with full-gene mutation, it has been proved in the literature [27] that single-gene mutation can dramatically improve the search efficiency of the algorithm. Denote r_z^t as a CQA in $A'_i(t) = \{r_{i1}(t), r_{i2}(t), \ldots, r_{iq_i-1}(t)\}$ which is generated by clone operation on the ith CQA of population $A(t)$. Choose the jth allele $\left[x_{zj}^t \;\; \alpha_{zj}^t \;\; \beta_{zj}^t\right]^T$ randomly from r_z^t and perform two kinds of Gaussian mutation on it:

$$\begin{aligned} x_{zj}^{t+1,\omega} &= N\left(\mu_{zj}^{t,\omega}, \left(\delta_{zj}^{t,\omega}\right)^2\right) \\ &= \begin{cases} x_{zj}^t + N\left(0, \left|\alpha_{zj}^t\right|^2\right), & \omega = \alpha, \\ \dfrac{x_{\max} - x_{\min}}{2} + N\left(0, \dfrac{\left|\beta_{zj}^t\right|^2}{\chi}\right), & \omega = \beta. \end{cases} \end{aligned} \tag{15}$$

χ is a variable. After the Gaussian mutation, $x_{zj}^{t+1,\omega}$ might exceed the interval $[0, 1)$. To avoid it, redefine $x_{zj}^{t+1,\omega}$ as follows:

$$x_{zj}^{t+1,\omega} = \begin{cases} \dfrac{k-1}{k}, & x_{zj}^{t+1,\omega} > 1, \\ 0, & x_{zj}^{t+1,\omega} < 0, \\ N\left(\mu_{zj}^{t,\omega}, \left(\delta_{zj}^{t,\omega}\right)^2\right), & \text{others.} \end{cases} \tag{16}$$

Gaussian mutation consists of two operations: one performs a local search around the current solution with a variance of $\left|\alpha_{zj}^t\right|^2$, another performs a wide-range search around the mean value with a variance of $\left|\beta_{zj}^t\right|^2/\chi$ lest the algorithm converges to a local optimal solution.

After the Gaussian mutation indicated by (15) and (16), the algorithm will calculate the *channel affinity* of the new antibody and compare it with the original one, that is, to decide which one is better between the two feasible solutions $(x_{z0}^{t+1}, \ldots, x_{zj}^{t+1,\omega}, \ldots, x_{z,m-1}^{t+1})$ and $(x_{z0}^t, \ldots, x_{zj}^t, \ldots, x_{z,q_i-1}^t)$. If the Gaussian mutation does improve the quality of the antibody, replace $x_{zj}^{t,\omega}$ with $x_{zj}^{t+1,\omega}$, and keep the probability of

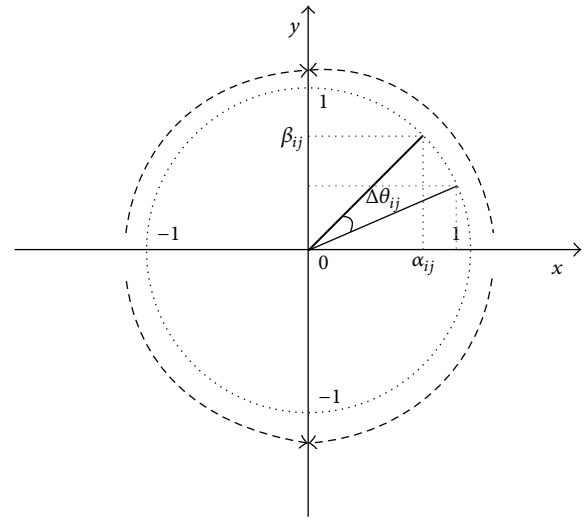


Figure 2: Sketch map of QRD operation.

$x_{zj}^{t,\omega}$ unchanged, that is, $\alpha_{zj}^{t+1} = \alpha_{zj}^t$, $\beta_{zj}^{t+1} = \beta_{zj}^t$. Otherwise, the Gaussian mutation has no effect and the probability indicating that the current solution is optimal should be increased. Quantum Rotation Door (QRD) is adopted to update $\left|\alpha_{zj}^t\right|^2$ and $\left|\beta_{zj}^t\right|^2$: the former will be decreased and the latter will be correspondingly increased after the rotation operation. Assume that the step size of the rotation is $\Delta\theta_{zj}^t$, the newly generated α_{zj}^{t+1} and β_{zj}^{t+1} are decided by (17)

$$\begin{bmatrix} \alpha_{zj}^{t+1} \\ \beta_{zj}^{t+1} \end{bmatrix} = \begin{bmatrix} \cos\left(\Delta\theta_{zj}^t\right) & -\sin\left(\Delta\theta_{zj}^t\right) \\ \sin\left(\Delta\theta_{zj}^t\right) & \cos\left(\Delta\theta_{zj}^t\right) \end{bmatrix} \begin{bmatrix} \alpha_{zj}^t \\ \beta_{zj}^t \end{bmatrix}. \tag{17}$$

During the evolution, the probability of composite vector $(\alpha_{zj}^t, \beta_{zj}^t)$ will approach gradually towards y-axis, shown as in Figure 2:

(1) If $(\alpha_{zj}^t, \beta_{zj}^t)$ lies in the first or the third quadrant, an anticlockwise rotation is needed and $\Delta\theta_{zj}^t$ is positive.

(2) If $(\alpha_{zj}^t, \beta_{zj}^t)$ lies in the second or the fourth quadrant, a clockwise rotation is needed and $\Delta\theta_{zj}^t$ is negative.

(3) If $(\alpha_{zj}^t, \beta_{zj}^t)$ lies exactly on y-axis, the algorithm has converged and the current feasible solution is the optimal one.

To sum up, $\Delta\theta_{zj}^t$ is realized as follows:

$$\Delta\theta_{zj}^t = \operatorname{sgn}\left(\alpha_{zj}^t \beta_{zj}^t\right) \Delta\theta \exp\left(-\frac{\left|\beta_{zj}^t\right|^2}{\left|\alpha_{zj}^t\right|^2 + N_q/N_s}\right), \tag{18}$$

where $\operatorname{sgn}(\cdot)$ is a sign function used to control the direction of the rotation, thus to make sure that the algorithm will finally converge to an optimal solution. $\Delta\theta$ means the maximum rotation angle in a single rotation operation. Current probability $\left|\alpha_{zj}^t\right|^2$, $\left|\beta_{zj}^t\right|^2$ as well as N_q/N_s is employed to control

the rotation dimension, or that is to say, to adjust the evolutionary speed in case of precocity of an antibody: N_q represents the total number of CQAs in $A'(t)$, that is, $N_q = \sum_{i=1}^{N} q_i$ and N_s indicates how stable the current population $A(t)$ is. After every operation, including Gaussian mutation and crossover, which may update x_{ij}^t, N_s should be recalculate as follows:

$$N_s = \begin{cases} N_s + 1, & x_{ij}^{t+1} = x_{ij}^t,\ i = 1, 2, \ldots, N, \\ & j = 0, 1, \ldots, m-1, \\ 1, & x_{ij}^{t+1} \neq x_{ij}^t,\ i = 1, 2, \ldots, N, \\ & j = 0, 1, \ldots, m-1. \end{cases} \tag{19}$$

4.2.3. Immune Selecting Operation. There shall form a new population:

$$A''(t) = \left\{A(t), A_1''(t), \ldots, A_N''(t)\right\} \tag{20}$$

after the immune genetic variation wherein

$$A_i''(t) = \left\{r_{i1}''(t), r_{i2}''(t), \ldots, r_{iq_i-1}''(t)\right\}, \quad i = 1, 2, \ldots, N. \tag{21}$$

For each CQA in population $A''(t)$ derived from immune manipulation operation, we firstly map $X = [x_0, x_1, \ldots, x_{m-1}] \in R^m$ to channel selection scheme $\mathbf{C} = [c_{s_0}, c_{s_1}, \ldots, c_{s_{m-1}}] \in Z^m$. Then compare them with original r_{i0}'' in S_m on the basis of *channel affinity*. For all $i = 1, 2, \ldots, N$, if exists

$$\mathbf{C}_{b_i}(t) = \left\{\mathbf{C}_{r_{ij}''}(t) \mid \max Q\left(\mathbf{C}_{r_{ij}''}\right), j = 0, 1, 2, \ldots, q_i - 1\right\} \tag{22}$$

making $Q(\mathbf{C}_{r_i}) < Q(\mathbf{C}_{b_i})$, $i = 1, 2, \ldots, N$, then $r_{i0}''(t+1) = b_i(t)$, otherwise $r_{i0}''(t+1)$ remains unchanged as $r_{i0}''(t)$. Immune selection operation selects the optimal CQA from all the antibodies generated by clone operation and immune manipulation operation as well as the immune memory set S_m to form a brand new population. After the operation is down, we can get not only new immune memory set S_m but also the new generation of CQA population $A(t+1)$.

4.2.4. Full Interference Crossover. To make full use of the information of all CQAs in the population, thus to guarantee that new antibodies will be generated in case of antibody precocity, which may cause the algorithm converge to a local optimal solution, a full interference crossover strategy [28] is adopted in this paper. Denote the jth allele in the ith antibody before and after the crossover operation to be A_{ij} and B_{ij} respectively; the relationship between A_{ij} and B_{ij} can then be revealed as $B_{ij} = A_{[(i+j)\%N][j]}$. A simple example is shown in Tables 1(a) and 1(b) to help understand when the population scale is set to $N = 5$, and the number of the monitoring sensors and current available channels is set to $m = 8$ and $k = 5$, respectively.

Table 1: (a) Group information before the crossing. (b) Group information after the crossing.

(a)

NO.0:	A_{00}	A_{01}	A_{02}	A_{03}	A_{04}	A_{05}	A_{06}	A_{07}
NO.1:	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
NO.2:	A_{20}	A_{21}	A_{22}	A_{23}	A_{24}	A_{25}	A_{26}	A_{27}
NO.3:	A_{30}	A_{31}	A_{32}	A_{33}	A_{34}	A_{35}	A_{36}	A_{37}
NO.4:	A_{40}	A_{41}	A_{42}	A_{43}	A_{44}	A_{45}	A_{46}	A_{47}

(b)

NO.0:	A_{00}	A_{11}	A_{22}	A_{33}	A_{44}	A_{05}	A_{16}	A_{27}
NO.1:	A_{10}	A_{21}	A_{32}	A_{43}	A_{04}	A_{15}	A_{26}	A_{37}
NO.2:	A_{20}	A_{31}	A_{42}	A_{03}	A_{14}	A_{25}	A_{36}	A_{47}
NO.3:	A_{30}	A_{41}	A_{02}	A_{13}	A_{24}	A_{35}	A_{46}	A_{07}
NO.4:	A_{40}	A_{01}	A_{12}	A_{23}	A_{34}	A_{45}	A_{06}	A_{17}

4.3. Algorithm Description. Based on the discussion, the process of MQICA is described as follows.

Step 1. Set algorithm parameters and initialize population $A(0)$. Calculate the initial channel affinity of each CQA in the population, that is, the Quality of Monitoring (QoM).

Step 2. Calculate the clone scale of each CQA according to (12) and then execute clone operation. After this step, $A'(t)$ is obtained.

Step 3. Do mutation operation on $A'(t)$, and get $A''(t)$.

Step 4. Do immune selection, and those selected antibodies constitute the new population $A(t+1)$.

Step 5. Calculate the channel affinity of each CQA in the new population as well as the evolutionary entropy of the population: if the former does not change any more and the latter tends to be close to zero infinitely or $t > t_{\max}$, the algorithm has already approximately converged, otherwise, crossover operation is applied to $A(t+1)$ and jump to Step 2.

The pseudo code of the algorithm is also given in Pseudocode 1.

5. Performance Analysis of MQICA

We will firstly prove that MQICA, just like traditional QICA, has a dramatic ability on global optimization searching. And in next chapter, lots of experiments are given to further identify the outstanding performance of MQICA, especially on MQoM problems.

Lemma 6. *The population sequence $\{A_t, t \geq 0\}$, generated by the evolutionary process of MQICA, is a stochastic process with discrete parameters and constitutes a time homogeneous Markov chain.*

Proof. Suppose that the state space of a single CQA is Ω, $A = \{r_1, r_2, \ldots, r_N\}$ represents the population where r_i responds to

Input: the sniffer set $S = \{s_1, s_2, \ldots, s_m\}$, channel set $C = \{c_1, c_2, \ldots, c_k\}$, user set $U = \{u_1, u_2, \ldots, u_n\}$, $c(u_j)$ and p_{u_j} $(j = 1, 2, \ldots, n)$, the iterative terminal entropy ε and the maximum iterations $t_{\max}$.
Output: the channel allocation vector **C**.

(1) $r_i(t) \leftarrow$ Generate initial population, $1 \le i \le N$, $t = 0$;
(2) Do *mappings between antibody to channel*, calculate the *evolutionary entropy* of each CQA and get $H(X_{r_i})$;
(3) $Q_i(t) \leftarrow$ function($\mathbf{C}_{r_i}$); /*compute the *channel affinity* of each antibody according to (2)*/
(4) $r''_{i0}(t) \leftarrow r_i(t)$ /*copy the original message to S_m*/
(5) **do** {
(6) $A'(t) \leftarrow A(t)$; /* clone operation*/
(7) $A''(t) \leftarrow A'(t)$; /*immune genetic variation*/
(8) Update immune memory set S_m;
(9) /*immune selection operation*/
(10) $A(t+1) \leftarrow S_m$;
(11) $Q^* = \max(Q_i(t))$; /*choose the max value*/
(12) Calculate the *evolutionary entropy* update $H(X_{r_i})$;
(13) **if** $\left((t < t_{\max}) \mid \left(\sum_{i=1}^{N} H\left(X_{r_i}\right) > \varepsilon\right)\right)$ **then**
(14) $t = t + 1$;
(15) Do full interference cross and update $A(t)$;
(16) **end**;}
(17) **while** $\left((t < t_{\max}) \mid \left(\sum_{i=1}^{N} H\left(X_{r_i}\right) > \varepsilon\right)\right)$;

Pseudocode 1

the ith antibody in A and N is the population scale. $r_i \in \Omega$, $A \in \Omega^N$. During the evolution process, all CQAs are discrete. $X = \lfloor \mathbf{C}/k \rfloor \in \{0, 1/k, \ldots, (k-1)/k\}$ describes the channel selection scheme. If the quantity of the CQA in population A is m, $\Omega = \{0, 1/k, \ldots, (k-1)/k\}^m \notin \phi$. So the population state space should be $|\Omega^N| = k^{Nm}$, that is, the state space during the evolution is finite. According to literature [29], denote δ to be the minimum σ-*algebra* generated by all cylinder set of Ω and P to be a real-value measure function defined in (Ω, δ), and thus the probability space of an CQA can be expressed as (Ω, δ, P) and the probability space of MQICA should be defined as (Ω^N, δ, P). As a result, $\{A(t), t \ge 0\}$, defined in state space (Ω, δ, P), is a stochastic sequence with discrete parameter and will change with t, the evolutionary times of our algorithm. Obviously, when the state space is replaced by (Ω^N, δ, P), the conclusion previously mentioned still holds.

Furthermore, the operations adopted in MQICA, including clone (T_c), immune gene manipulation (T_m & T_e), and immune selection (T_I), guarantee that $A(t+1)$ is only related to $A(t)$. So $\{A(t), t \ge 0\}$ is time homogeneous Markov chain in state space (Ω^N, δ, P). □

Definition 7. Denote $\mathbf{a}^* = \mathbf{C}^*$ as the optimal channel selection scheme, namely, $\mathbf{C}^* = \arg\max Q(\mathbf{a}) = \arg Q^*$, where Q^* is the QoM value corresponding to optimal channel selection scheme. MQICA will converge to global optimal solution when and only when $\lim_{t \to \infty} P(Q_t = Q^*) = 1$.

Lemma 8. *The transition probability matrix $M(T_m)$, which indicates that the probability for a CQA in clone group $A'(t)$ changes its state from $\mathbf{\Lambda}^{\mu}$ to $\mathbf{\Lambda}^{\lambda}$ after the MQICA mutation operation T_m, is strictly positive.*

Proof. For Gaussian mutation operation G, shown in (15), assume that after the mutation $x_{zj}^{t+1} = \begin{cases} x_{zj}^{t+1,\alpha} \\ x_{zj}^{t+1,\beta} \end{cases}$, the probability for x_{zj}^{t} to mutate to $x_{zj}^{t+1,\alpha}$ should be

$$G\left(x_{z,j}^{t}, x_{z,j}^{t+1,\alpha}\right) = \int_{x_{z,j}^{t}}^{x_{z,j}^{t+1,\alpha}} \left(\frac{1}{\sqrt{2\pi}\left|\alpha_{z,j}^{t}\right|} \cdot \exp\left(-\frac{\left(x - x_{z,j}^{t+1,\alpha}\right)^2}{\pi\left|\alpha_{z,j}^{t}\right|^2} \right) \right) \cdot dx > 0. \tag{23}$$

By the same token, the probability for x_{zj}^{t} to mutate to $x_{zj}^{t+1,\beta}$ should be

$$G\left(x_{z,j}^{t}, x_{z,j}^{t+1,\beta}\right) = \int_{x_{z,j}^{t}}^{x_{z,j}^{t+1,\beta}} \left(\frac{1}{\sqrt{2\pi}\left|\beta_{z,j}^{t}\right|} \cdot \exp\left(-\frac{\left(x - x_{z,j}^{t+1,\beta}\right)^2}{\pi\left|\beta_{z,j}^{t}\right|^2} \right) \right) \cdot dx > 0. \tag{24}$$

Because these two Gaussian mutations are independent, so the probability of state transition from $\mathbf{x}_\mu$ to $\mathbf{x}_\lambda$ after this operation would be

$$\begin{aligned} G\left(\mathbf{x}_\mu, \mathbf{x}_\lambda\right) &= G\left(x_{z,j}^t, x_{z,j}^{t+1}\right) \\ &= G\left(x_{z,j}^t, x_{z,j}^{t+1,\alpha}\right) \cdot G\left(x_{z,j}^t, x_{z,j}^{t+1,\beta}\right) > 0. \end{aligned} \tag{25}$$

When the state of α in a specific allele of an antibody is changed from a_λ to a_μ by QRD operation, the state transition probability $U(a_\mu, a_\lambda) > 0$. Thus,

$$\begin{aligned} m_{\mu\lambda}\left(T_m\right) &= G\left(\mathbf{x}_\mu, \mathbf{x}_\lambda\right) \times \left(1 - U\left(a_\mu, a_\lambda\right)\right) \\ &\quad + \left(1 - G\left(\mathbf{x}_\mu, \mathbf{x}_\lambda\right)\right) \times U\left(a_\mu, a_\lambda\right) > 0, \end{aligned} \tag{26}$$

and obviously the transition probability matrix $M(T_m)$ is strictly positive. □

Lemma 9. *The state transition matrix P for MQICA is a regular one.*

Proof. The state transition process of the population in Ω^N is described by the following four operations: T_c, T_m, T_I, and T_e. Denote T to be $T = T_c \cdot T_m \cdot T_I \cdot T_e$, and as a result, $r_i(t+1) = T[r_i(t)] = T_c \cdot T_m \cdot T_I \cdot T_e[r_i(t)]$. Assume that the state of the population was transferred from $\mathbf{\Lambda}^\mu$ to $\mathbf{\Lambda}^\lambda$ after the tth iteration, where $\mathbf{\Lambda}^\mu, \mathbf{\Lambda}^\lambda \subseteq \Omega^N$. So the state transition probability of MQICA is

$$\begin{aligned} p_{\mu\lambda}(t) &= p\{A(t+1) = Y \mid A(t) = W\} = p(Y \mid W) \\ &= \prod_{i=1}^{N} \left\{ p\left\{T_c\left(r_i(t)\right)\right\} \times \prod_{j=1}^{q_i - 1} p\left\{T_m\left(r_{ij}(t)\right)\right\} \cdot p_m \right. \\ &\quad \times \prod_{\eta=1}^{t} p\left\{T_I\left(r_{\lambda i}(t+1) = r_{\mu i}(t)\right)\right\} \\ &\quad \left. \times p\left\{T_e\left(r_{\mu i}(t+1)\right)\right\} \cdot p_c \right\}, \end{aligned} \tag{27}$$

where,

$$\begin{aligned} p\left\{T_c\left(r_i(t)\right)\right\} &= \frac{Q\left(\mathbf{C}_i\right)}{\sum_{j=1}^{N} Q\left(\mathbf{C}_j\right)} > 0, \\ p\left\{T_m\left(r_{ij}(t)\right)\right\} &= m_{\mu\lambda}\left(T_m\right) \geq 0. \end{aligned} \tag{28}$$

Because the full interference cross has fixed relationships, that is, $p\{T_e(r_{\mu i}(t+1))\} = 1$, the lemma can be proved with the following three conditions.

Condition 1. When $Q(\mathbf{C}_{\mu i}) < Q(\mathbf{C}_{\lambda i})$,

$$p\left\{T_I\left(r_{\lambda i}(t+1) = r_{\mu i}(t)\right)\right\} = 1. \tag{29}$$

Condition 2. When $Q(\mathbf{C}_{\mu i}) > Q(\mathbf{C}_{\lambda i})$,

$$p\left\{T_I\left(r_{\lambda i}(t+1) = r_{\mu i}(t)\right)\right\} = 0. \tag{30}$$

Condition 3. When $Q(\mathbf{C}_{\mu i}) = Q(\mathbf{C}_{\lambda i})$, the evolutionary entropy of the CQA satisfies $H(X) \to 0$; in other words, the algorithm should be converged and $p_{\mu\lambda}(t) = p_{\mu\mu}(t) = 1$, and the state transition probability of MQICA can be summarized as

$$\begin{aligned} p_{\mu\lambda}(t) > 0, &\quad \text{s.t. } Q\left(\mathbf{C}_{\mu i}\right) \leq Q\left(\mathbf{C}_{\lambda i}\right), \\ p_{\mu\lambda}(t) = 0, &\quad \text{s.t. } Q\left(\mathbf{C}_{\mu i}\right) > Q\left(\mathbf{C}_{\lambda i}\right). \end{aligned} \tag{31}$$

Obviously, $P \geq 0$ and $\exists t$ that makes $P^t > 0$. Thus P is a is regular matrix. □

Lemma 10. *The Markov chain derived from MQICA is ergodic.*

Proof. Lemma 9 indicates that the state transition matrix P for MQICA is regular, and because the Markov cycle is 1, based on the basic Markov limit theorem, a unique limit $\lim_{t\to\infty} P^t = P^*$ must exist. Because $P^* > 0$, so the homogeneous Markov chain is nonzero and recurrent, thus any state in this chain would have an only limit distribution with a probability that is greater than zero regardless of how the population is initialized. As a result, MQICA can start from state i to state j within limited time; that is, when $t \to \infty$, this Markov chain could traverse the whole state space. □

Lemma 11. *MQICA converges to the global optimal solution on a probability of 1.*

Proof. MQICA adopts a so-called survival of the fittest strategy, which means that the channel affinities of this Markov sequence, generated by the evolution, are monotone and will not decrease. $\mathbf{\Lambda}^* \subseteq \Omega^N$ represents the population containing the global optimal antibody $\mathbf{r}^*$. $\mathbf{C}^*$ denotes the global optimal channel selection scheme, while Q^* denotes the global optimal channel affinity. Since the evolution process would not degenerate, $\mathbf{\Lambda}^*$ is a closed set and will be always in an attractive state, which means that for all $i \in \mathbf{\Lambda}^*$, $\sum_{j\in\mathbf{\Lambda}^*} p_{ij} = 1$ always holds. So once the state of the population A_t is changed to $\mathbf{\Lambda}^*$, there would be no chance for the population to enter other state.

Based on the basic Markov limit theorem, MQICA will definitely reach state $\mathbf{\Lambda}^*$ after limited steps t_c if only the state transition matrix P is regular and the corresponding Markov chain is ergodic, which have been proved by Lemmas 9 and 10. Thus, the following equation is satisfied:

$$\lim_{t\to\infty} P\left(Q(t) = Q_{t_c} = Q^*\right) = 1. \tag{32}$$

Table 2: Parameters setting.

N	p_m	p_c	N_c	χ	ε	$t_{\max}$
10	0.5	0.88	15	3	10^{-5}	1000

This indicates that MQICA converges to the global optimal solution on a probability of 1. □

6. Experiment Results

6.1. Simulations. In this paper, we conduct extensive experiments to validate the effectiveness of the algorithm. The program is run on a PC with Intel(R) Core(TM)2 CPU @2.40 GHz, 2 GB memory. The software platform is Windows XP. Table 2 lists the parameters of MQICA.

N is the population scales. Large N can promote the searching ability of the algorithm, meanwhile extend the running time of program. The other parameters are all set as the experiential value for MQICA applications, and the experiments result also shows the validity in this case.

From Section 5 we have known the validity of the proposed algorithm, MQICA, in solving multichannel allocation problems. Now a mass of experiment results also elaborate the effectiveness in another way. Firstly, we tentatively do three different experiments 5 times, respectively, according to the size. For small scale, $m = 3$, $k = 2$, $n = 25$; for medium scale, $m = 12$, $k = 6$, $n = 200$; for large scale, $m = 12$, $k = 9$, $n = 1000$. The experiment results are shown in Figures 3(a), 3(b), and 3(c). As can be seen from the graph, no matter how initialization is, MQICA will eventually well converge to the same optimal solution.

Secondly, in order to validate the correctness of the algorithm and eliminate the possibility of local optimal solution, we take traversal method for the small scale monitoring network. Ergodic results are shown as follows: 1.1, 1.1, 0.8, 0.767, 1.15, 1.15, 0.767, and 0.75. Obviously, MQICA can quickly find the optimal solution in small time. For medium and large scale, we both do the test fifty times. The results are expressed in Tables 3 and 4.

From Table 3, During 50-times experiment, we can see that initial channel scheme is random so the initial QoM value is not optimal. But after a certain number of iterations, the network monitoring quality has been converged to or close to the optimal value of 9.345. Similarly, Table 4 shows that the algorithm can still do a better performance for the distribution of channel options under lager networks.

Now we can easily conclude that MQICA will generate a good performance in channel allocation problems. It can be quickly uniform convergence to the optimal solution when the size of monitoring networks is small or moderate. If the scale is large; MQICA can also be better converged to the optimal or near optimal solution in most cases. These experimental results have proved the effectiveness of the proposed algorithm from various scales.

We also evaluate the performance of MQICA comparing three baseline algorithms.

Table 3: Result for medium scale.

ID	Q_0	Q^*	Times
1	2.439	9.345	473
2	2.536	9.345	431
3	6.304	9.345	487
4	9.003	9.345	375
5	8.604	9.345	510
6	6.966	9.345	457
7	7.444	9.171	218
8	8.046	9.345	324
9	8.667	9.345	321
10	8.856	9.345	78
11	8.871	9.345	310
12	2.122	9.345	454
13	7.164	9.345	356
14	2.439	9.345	409
15	6.304	9.345	326
16	2.536	9.345	245
17	8.667	9.345	265
18	4.350	9.345	351
19	9.345	9.345	0
20	2.122	9.345	456
21	7.164	9.345	328
22	8.604	9.345	91
23	9.003	9.345	103
24	5.438	9.345	501
25	2.536	9.171	365
26	5.473	9.345	141
27	3.180	9.345	265
28	2.439	9.345	454
29	8.992	9.345	263
30	8.646	9.003	453
31	3.857	9.345	261
32	2.122	9.345	356
33	7.444	9.345	452
34	2.597	9.345	365
35	2.122	9.345	532
36	8.367	9.345	269
37	6.304	9.345	462
38	7.146	9.345	254
39	8.171	9.345	56
40	3.200	9.345	495
41	9.003	9.345	321
42	5.251	9.345	256
43	3.062	9.345	518
44	6.304	9.345	265
45	6.996	9.345	348
46	8.046	9.345	206
47	2.536	9.345	527
48	9.345	9.345	0
49	7.164	9.345	215
50	8.171	9.345	256

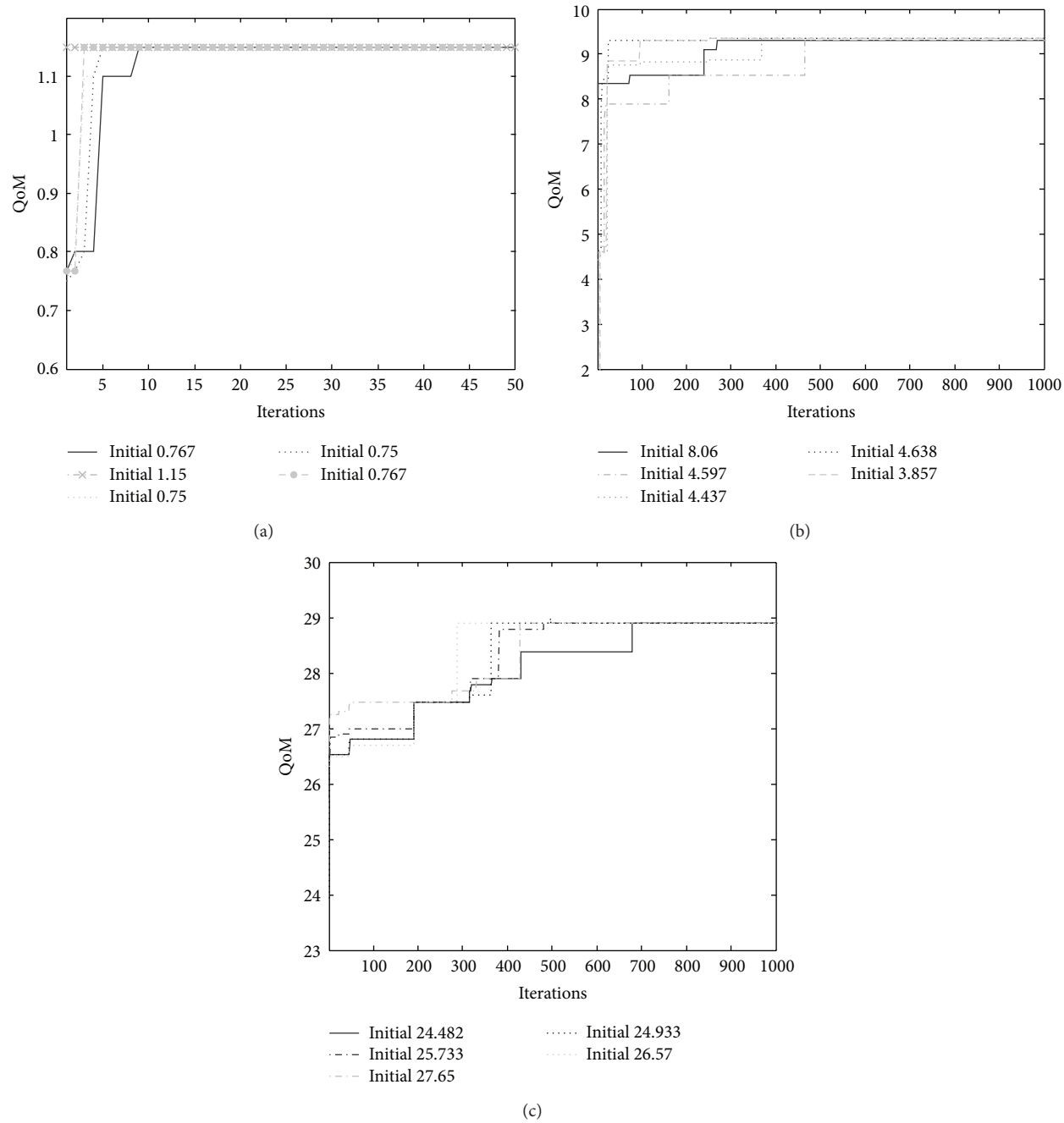


Figure 3: (a) The convergence of small scale. (b) The convergence of medium scale. (c) The convergence of large scale.

Greedy. Select channel for each sniffer to maximize the sum of transmission probability of its neighbor users.

Linear Programming (*LP*). Solve the integer programming problem from formula (4).

Gibbs Sampler. Sniffer computes the local energy of optional channels and their selection probability, then chooses one channel according to the probability.

We conducted four sets of experiments, and the number of optional channels is 2, 6, and 9, respectively. In each experiment, the four algorithms are compared in different aspects of performance. The algorithm program runs 30 times to get the average result for evaluation.

In the first set of experiment, 1000 users are distributed in $500 \times 500\,\mathrm{m}^2$ square field as shown in Figure 4; transmission probability $p_u \in [0, 0.06]$. The field is partitioned in several regular hexagon units to construct cellular framework. Each unit center is equipped with a base station (BS) working on a certain channel and users in the unit work on the same channel as BS. Every two adjacent units are on different channels. For easy to control, 25 sniffers are deployed

Table 4: Results for large scale.

ID	Q_0	Q^*	Times
1	24.150	28.900	684
2	26.367	28.900	596
3	27.200	28.850	352
4	28.900	28.900	0
5	28.250	28.900	152
6	25.850	29.050	715
7	27.417	28.900	254
8	24.533	28.900	561
9	28.450	28.900	215
10	27.250	28.900	325
11	27.200	28.900	387
12	28.467	28.900	152
13	24.533	28.900	356
14	27.050	28.900	261
15	28.700	28.900	164
16	28.150	28.850	92
17	26.900	28.900	364
18	24.533	28.900	259
19	24.750	28.900	381
20	25.850	28.900	681
21	28.267	28.900	154
22	24.750	29.050	296
23	27.200	28.900	614
24	25.700	28.900	265
25	24.150	28.900	562
26	28.567	28.900	244
27	27.800	29.050	264
28	25.700	28.900	157
29	24.533	28.900	246
30	27.800	28.900	106
31	27.417	28.900	315
32	28.850	28.900	26
33	27.800	29.050	268
34	24.533	28.900	654
35	27.200	28.900	341
36	25.700	28.900	465
37	26.750	28.900	287
38	28.467	28.900	384
39	27.900	29.050	468
40	27.200	28.900	215
41	24.750	28.900	216
42	24.150	28.900	656
43	23.587	28.900	146
44	25.700	28.900	356
45	28.267	28.850	21
46	27.200	28.900	265
47	24.533	28.900	378
48	27.800	28.900	198
49	28.700	28.850	254
50	26.750	28.900	394

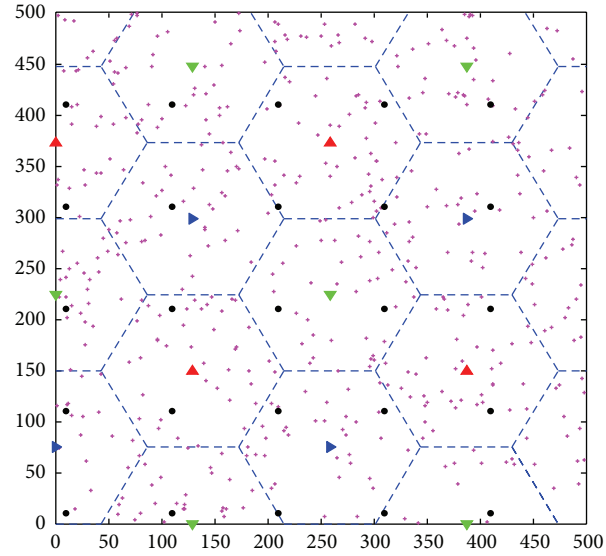


Figure 4: Wireless network topology Hexagonal layout with users (purple "+"), sniffers (solid dots), and base stations (isosceles triangles) of each cell (different color representing working on different channels).

uniformly in the field to form a network to monitor the communication activities of users in the field. Monitoring radius of sniffer is 120 meters and 3 optional channels (in IEEE 802.11.b/g WLAN, there are 3 orthogonal channels, the 1st, 6th, and 11th, with center frequency 2412 MHz, 2437 MHz, and 2462 MHz). MQICA, LP, Greedy, and Gibbs Sampler are applied separately to solve the optimal channel selection scheme for sniffers. The quality of solution (QoM) of the four algorithms is shown in Figure 5.

As depicted in Figure 5, after 700 iterations, the proposed MQICA algorithm converges to the extremely optimal solution (QoM = 28.975). LP algorithm takes the second place with QoM up to 28.105, while Gibbs Sampler and Greedy algorithm achieve the QoM of 27.048 and 23.893, respectively. It is shown that the Multiple Quantum Immune Clone Algorithm improves the convergence rate of other algorithm effectively and produces a better global searching ability.

Table 5 demonstrates the statistical results of the three sets of experiments. Among the four algorithms, MQICA and Gibbs Sampler both run 20 times in each set of experiments to get the average optimal solution and its QoM value. As deterministic methods, LP and Greedy just run once. From Table 5, we can see that MQICA outperforms LP in three sets of experiments and evidently better than Gibbs Sampler and Greedy. Furthermore, MQICA converges fast, with shorter running time than Gibbs Sample.

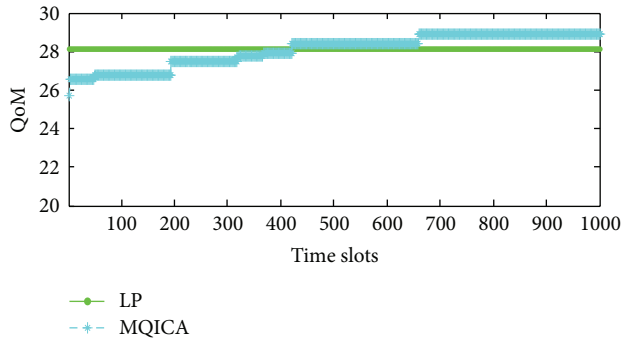
6.2. Practical Network Experiment. In this section, we evaluate the proposed MQICA algorithm by practical network experiment based on campus wireless network (IEEE 802.11.b WLAN). 21 WiFi sniffers were deployed in a building to collect the user information from 1 pm to 6 pm (over 5 hours).

Table 5: Statistical results of three sets of experiments.

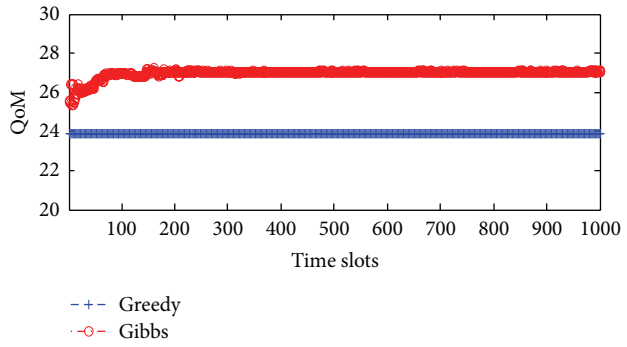
Experiment no.	MQICA		Gibbs sampler		LP		Greedy	
	Average optimal QoM	Running time/s (1000 iter.)	Average optimal QoM	Running time/s (1000 iter.)	QoM	Running time/s (1 iter.)	QoM	Running time/s (1 iter.)
1 (2 channels)	27.338	10.616	26.052	28.938	27.105	0.562	22.872	0.093
2 (6 channels)	26.760	11.695	26.261	30.953	26.484	0.812	23.363	0.109
3 (9 channels)	26.263	11.759	26.140	35.031	26.088	0.934	23.481	0.119

Table 6: Parameters setting.

Active probability	0~0.01	0.01~0.02	0.02~0.04
Number of users	578	15	29



(a)



(b)

Figure 5: Performance comparison of the four algorithms in the first set of experiments (3 optional channels).

Each sniffer captured approximately 320,000 MAC frames. Totally 622 users were monitored working on 3 orthogonal channels. The number of users in 1st, 2nd, and 3rd channels is 349, 118, and 155, respectively. The activity probabilities (active probability of a user is computed as the percentage of the user's active time in a unit time.) of these users were recorded in Table 6. It is shown that the activity probabilities of most users are less than 1%. The average activity probability is 0.0026.

Figure 6 depicted the QoM of network with different number of sniffers. It is clear that the QoM (the number of monitored active users) is growing up with the increment of sniffers (from 5 to 21). Except the experiment with 21 sniffers, the other sets of experiments were conducted repeatedly with different sniffers selected randomly from the 21 sniffers, and the statistical average value of QoM was recorded and shown in Figure 6. Since the average activity probability is 0.0026, the largest number of active users is less than 1.7 during every time slot. By comparing with LP, Gibbs Sampler, and Greedy, the proposed MQICA exhibits its superiority and feasibility in the practical network environment.

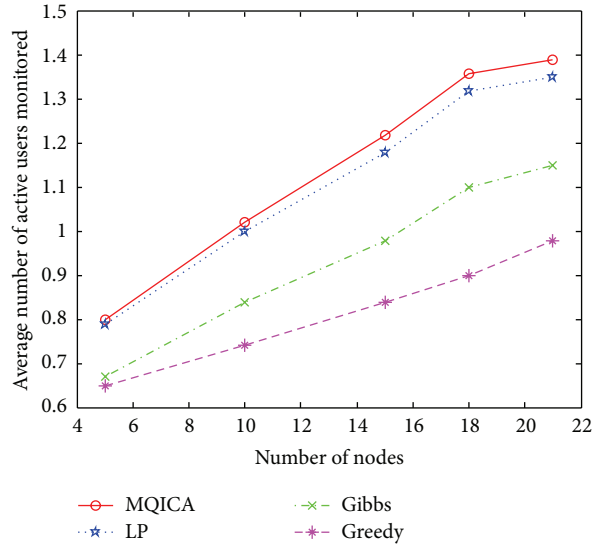


Figure 6: QoM of campus wireless network with different number of sniffers.

7. Conclusion

In this paper, we investigate the channel allocation for sniffers to maximize the Quality of Monitoring (QoM) for wireless monitoring networks, which is proved to be NP-hard. A Multiple-Quantum-Immune-Clone-based channel selection algorithm (MQICA) is put forward to solve the problem. By theoretical proof and extensive experiments, we demonstrate that MQICA can solve the channel allocation problem effectively, and outperform related algorithms evidently with fast convergence. As an ongoing work, we are reducing the computation complexity and proving the convergence performance of algorithm in theory.

Acknowledgments

This work is funded by the National Science Foundation of USA (CNS-0832089), the National Natural Science Fund of China (61100211 and 61003307), and the Postdoctoral Science Foundation of China (20110490084 and 2012T50569).

References

[1] J. Zander, S. L. Kim, and M. Almgren, *Radio Resource Management for Wireless Networks*, Artech House, Norwood, Mass, USA, 2001.

[2] L. M. Correia, D. Zeller, O. Blume et al., "Challenges and enabling technologies for energy aware mobile radio networks," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 66–72, 2010.

[3] Y. Liu, K. Liu, and M. Li, "Passive diagnosis for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1132–1144, 2010.

[4] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless LAN monitoring and its applications," in *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe '04)*, pp. 70–79, ACM, New York, NY, USA, October 2004.

[5] J. Yeo, M. Youssef, T. Henderson, and A. Agrawala, "An accurate technique for measuring the wireless side of wireless networks," in *Proceeding of the Workshop on Wireless Traffic Measurements and Modeling (WiTMeMo '05)*, pp. 13–18, USENIX Association, Berkeley, Calif, USA, 2005.

[6] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based characterization of 802.11 in a hotspot setting," in *Proceedings of the ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, pp. 5–10, New York, NY, USA, August 2005.

[7] Y. C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Solving the puzzle of enterprise 802. 11 analysis," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '06)*, pp. 39–50, ACM, New York, NY, USA, 2006.

[8] W. G. Yang, T. D. Guo, and T. Zhao, "Optimal lifetime model and it's solution of a heterogeneous surveillance sensor network," *Chinese Journal of Computers*, vol. 30, no. 4, pp. 532–538, 2007.

[9] C. Liu and G. Cao, "Distributed monitoring and aggregation in wireless sensor networks," in *Proceedings of the 29th Conference on Information Communications (INFOCOM '10)*, pp. 1–9, IEEE Press, Piscataway, NJ, USA, 2010.

[10] J. Jin, B. Zhao, and H. Zhou, "DLDCA: a distributed link-weighted and distance-constrained channel assignment for single-radio multi-channel wireless mesh networks," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '09)*, pp. 1–5, Nanjing, China, November 2009.

[11] C. E. A. Campbell, K. K. Loo, and R. Comley, "A new MAC solution for multi-channel single radio in wireless sensor networks," in *Proceedings of the 7th International Symposium on Wireless Communication Systems (ISWCS '10)*, pp. 907–911, York, UK, September 2010.

[12] Z. Zhang and X. Yu, "A simple single radio multi-channel protocol for wireless mesh networks," in *Proceedings of the 2nd International Conference on Future Computer and Communication (ICFCC '10)*, pp. V3-441–V3-445, Wuhan, China, May 2010.

[13] D. H. Shin and S. Bagchi, "Optimal monitoring in multi-channel multi-radio wireless mesh networks," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 229–238, MobiHoc, New Orleans, La, USA, May 2009.

[14] B. J. Kim and K. K. Leung, "Frequency assignment for IEEE 802.11 wireless networks," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, pp. 1422–1426, Orlando, Fla, USA, October 2003.

[15] C. Chekuri and A. Kumar, "Maximum coverage problem with group budget constraints and applications," in *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pp. 72–83, APPROX, Cambridge, Mass, USA, 2004.

[16] A. Chhetri, H. Nguyen, G. Scalosub, and R. Zheng, "On quality of monitoring for multi-channel wireless infrastructure networks," in *Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '10)*, pp. 111–120, MobiHoc, Chicago, Ill, USA, September 2010.

[17] P. Arora, N. Xia, and R. Zheng, "A gibbs sampler approach for optimal distributed monitoring of multi-channel wireless networks," in *Proceedings of IEEE GLOBECOM*, pp. 1–6, IEEE Communication Society Press, 2011.

[18] I. Wormsbecker and C. Williamson, "On channel selection strategies for multi-channel MAC protocols in wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '06)*, pp. 212–220, IEEE Computer Society, Washington, DC, USA, June 2006.

[19] C. N. Kanthi and B. N. Jain, "Spanner based distributed channel assignment in wireless mesh networks," in *Proceedings of the 2nd International Conference on Communication System Software and Middleware and Workshops (COMSWARE '07)*, pp. 1–10, COMSWARE, Bangalore, India, January 2007.

[20] L. You, P. Wu, M. Song, J. Song, and Y. Zhang, "Dynamic control and resource allocation in wireless-infrastructured distributed cellular networks with OFDMA," in *Proceedings of the 38th International Conference on Parallel Processing Workshops (ICPPW '09)*, pp. 337–343, Vienna, Austria, September 2009.

[21] F. Hou and J. Huang, "Dynamic channel selection in cognitive radio network with channel heterogeneity," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, pp. 6–12, Miami, Fla, USA, December 2010.

[22] Z. G. Du, P. L. Hong, W. Y. Zhou, and K. P. Xue, "ICCA: Interface-clustered channel assignment in multi-radio wireless mesh networks," *Chinese Journal of Electronics*, vol. 39, no. 3, pp. 723–726, 2011.

[23] L. Jiao and Y. Li, "Quantum-inspired immune clonal optimization," in *Proceedings of the International Conference on Neural Networks and Brain Proceedings (ICNNB '05)*, pp. 461–466, Beijing, China, October 2005.

[24] Y. Y. Li and L. C. Jiao, "Quantum-inspired immune clonal algorithm for SAT problem," *Chinese Journal of Computers*, vol. 30, no. 2, pp. 176–183, 2007.

[25] K. A. de Jong, *An analysis of the behavior of a class of genetic adaptive systems [Ph.D. thesis]*, University of Michigan, 1975.

[26] Y. Yu and C. Z. Hou, "A clonal selection algorithm by using learning operator," in *Proceedings of International Conference on Machine Learning and Cybernetics*, pp. 2924–2929, August 2004.

[27] X. Z. Wang and S. Y. Yu, "Improved evolution strategies for high-dimensional optimization," *Control Theory and Applications*, vol. 23, no. 1, pp. 148–151, 2006.

[28] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '96)*, pp. 61–66, Nogaya, Japan, May 1996.

[29] W. X. Zhang and Y. Liang, *The Mathematical Basis of Genetic Algorithm*, Press of Xi'an Jiaotong University, 2000.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 241261, 14 pages
http://dx.doi.org/10.1155/2013/241261

Research Article

ID List Forwarding Free Confidentiality Preserving Data Aggregation for Wireless Sensor Networks

Liehuang Zhu, Zhen Yang, Mingzhong Wang, and Meng Li

Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Liehuang Zhu; liehuangz@bit.edu.cn

Received 10 January 2013; Accepted 26 March 2013

Academic Editor: Xu Yongjun

Wireless sensor networks (WSNs) are composed of sensor nodes with limited energy which is difficult to replenish. Data aggregation is considered to help reduce communication overhead with in-network processing, thus minimizing energy consumption and maximizing network lifetime. Meanwhile, it comes with challenges for data confidentiality protection. Many existing confidentiality preserving aggregation protocols have to transfer list of sensors' ID for base station to explicitly tell which sensor nodes have actually provided measurement. However, forwarding a large number of node IDs brings overwhelming extra communication overhead. In this paper, we propose provably secure aggregation scheme perturbation-based efficient confidentiality preserving protocol (PEC2P) that allows efficient aggregation of perturbed data without transferring any ID information. In general, environmental data is confined to a certain range; hence, we utilize this feature and design an algorithm to help powerful base station retrieve the ID of reporting nodes. We analyze the accuracy of PEC2P and conclude that base station can retrieve the sum of environmental data with an overwhelming probability. We also prove that PEC2P is CPA secure by security reduction. Experiment results demonstrate that PEC2P significantly reduces node congestion (especially for the root node) during aggregation process in comparison with the existing protocols.

1. Introduction

Wireless sensor networks (WSNs) integrate microelectromechanical systems (MEMSs) technology, sensor technology, and communication technology. WSN can sense, transport and process different environmental data in its deployment area by hundreds of sensor nodes with limited computation and energy capacities. WSNs have been extensively used in military surveillance, environmental monitoring, production control, and real-time traffic monitoring [1].

Because WSNs are usually deployed in remote, unattended, or even hostile environment, the energy of sensor nodes is not easy to get replenished. Hence, how to reduce energy cost and prolong the network lifetime has become key issues for WSNs [2, 3]. It is generally believed that power consumption of each sensor node tends to be dominated by data transmission. According to [4], energy cost of transmitting a single bit of data is equivalent to that of 800 instructions. Data aggregation [2, 5] mechanisms avoid transmitting environmental data through in-network process of summarizing and combining sensor data, thus reducing the amount of data transmission and effectively maximizing network lifetime.

Data confidentiality [6–11] is crucial in many WSN applications, like military surveillance. If data confidentiality is compromised, the sensitive information collected will be leaked to adversary. However, there is a conflict between data aggregation and data confidentiality protocols [12]: data aggregation prefers to operate on plain data and confidentiality protection requires data to be encrypted. Extensive secure data aggregation research [6–11, 13–15] has been conducted. Data aggregation protocols usually cannot operate on encrypted data such that intermediate node has to decrypt packets received from downstream, aggregate the plaintext data with its own, encrypt the aggregated result, and forward to upstream.

Two common approaches to preserve data confidentiality without decryption/encryption are homomorphic

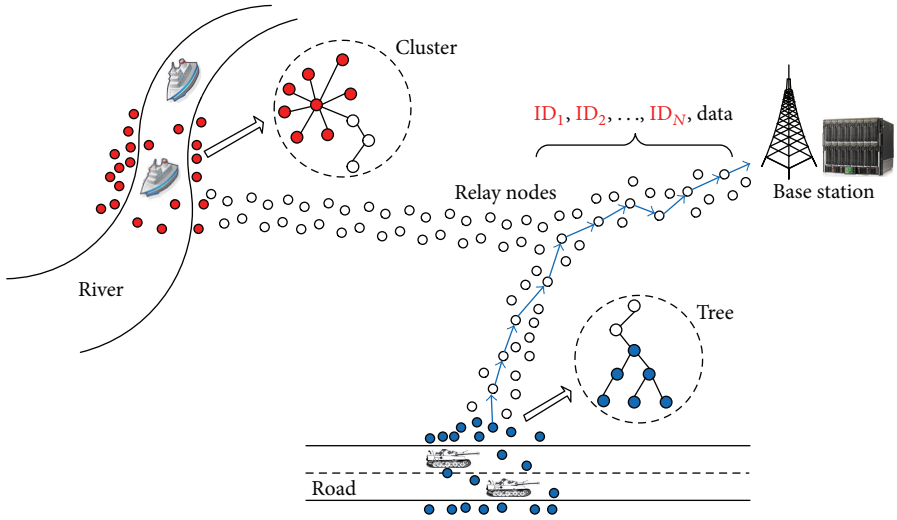


Figure 1: An example of environmental surveillance system in battlefield.

encryption [7, 8, 10], and secret perturbation [6, 9, 11]. Homomorphic encryption is an encryption transformation that allows direct computation on encrypted data. However, end-to-end security of symmetric homomorphism [7] is easily compromised if any node is corrupted and the computational cost and communication overhead of asymmetric homomorphism [8, 10] are not preferable. In comparison, secret perturbation-based schemes add a perturbation to the value of each reporting node using shared secret key with base station (*BS*). *BS* retrieves the final aggregation result by removing all these perturbation. Since the key shared between each node and *BS* is unique, adversary will not compute other nodes' sensed data or intermediate aggregation result if one key is compromised.

BS has to know which sensor nodes have provided measurement before it can correctly remove the perturbations brought by these sensor nodes. A straightforward solution is to require every node participating/not participating in aggregation process to report its ID, according to the proportion of nodes satisfying *BS*'s query.

However, this approach may bring high extra overhead. Feng et al. [9] proposed a family of secret perturbation-based schemes that can protect sensor data confidentiality while trying to minimize the number of ID to be transferred. In FSP scheme, every sensor node must reply a perturbed actual or dummy data item, no matter the node has satisfying data or not. *BS* will simply subtract hash value for every sensor node to compute final aggregation result, and communication overhead caused by ID transmission is avoided. However, it requires all sensor nodes to report data no matter whether they have data satisfying the query. This may result in high extra communication overhead when only a small number of sensor nodes have data to report and communication overhead caused by extra perturbed data can be much larger than that of forwarding ID. Hence, in their ideal scheme O-ASP, aggregating node first has to compute whether overhead of transmitting ID and perturbed data or overhead of transmitting all perturbed data is larger. Either way, O-ASP endures high communication overhead, and it is unrealistic for each sensor node to know the membership and topology of the whole network, and it knows whether each of these nodes has data satisfying each particular query.

Moreover, the transmission of nodes' ID makes [9, 11] not suitable for the scenario shown in Figure 1, where we want to monitor the activities of tanks and battleships, and there is a long path to travel through before aggregation result gets to *BS*. To achieve this goal, a cluster or a tree of sensor nodes is deployed in the battlefield, while *BS* is in a secure location away to collect data reported by sensors. All data has to be forwarded on a long path from the targeted area to *BS*. For [9, 11], ID list is transmitted such that the energy is wasted on the long path, and "single point of failure" could happen if there are not enough nodes on such path. The application scenarios in military surveillance also include the case that US army uses REMBASS to collect data (like ground motion, sound, infrareds, and magnetic fields) and forward the aggregation result to command center. PEC2P fits in this scenario and does not have any requirement on the type of data.

In this paper, we present perturbation-based efficient confidentiality preserving protocol (PEC2P) which can protect data confidentiality without transmitting any ID information. Generally, we use one-way hash function as perturbation added to the environmental data. Since *BS* usually has powerful computational capability in WSNs, we propose to trade computation consumption at *BS* for energy cost of sensors and introduce a new approach for *BS* to compute and tell which nodes have actually sensed data and contributed to the aggregation process after receiving the final aggregation result. Our approach specifically fits for scenarios where aggregation result has to travel a long path before arriving at *BS*. In summary, contribution of this paper includes the following.

(1) We draw attention to the ID-list transmitting problem in WSNs and propose the first approach which does

not require forwarding any node ID but computing and selecting by *BS*. As a result, communication overhead is reduced and reporting nodes' information is further hidden.

(2) We avoid using the random number r verified by commonly applied authenticated broadcasting, thus reducing network delay. Instead, we update the secret key of all reporting nodes after each data aggregation to keep indistinguishability from adversary.

(3) We prove that our protocol is CPA secure by security reduction.

(4) We measure the performance of our protocol through both theoretic analysis and experiments on TinyOS [16]. We analyze the accuracy of PEC2P and compare its communication overhead with existing protocols.

2. Related Work

Girao et al. proposed CDA [7] using symmetric key-based privacy homomorphic encryption. In their approach, sensor nodes share a common symmetric key with the *BS* which is hidden from aggregators, and aggregators can perform aggregation functions directly on the ciphertext instead of carrying out costly decryption and encryption operations. Symmetric homomorphism has the advantage of fast computation. However, secret key is shared among all nodes such that data confidentiality is lost once a sensor node and its shared key are compromised.

Mykletun et al. [8] investigated several additive homomorphic public-key encryption schemes and their applicability to WSNs. In general, these schemes preload public key in sensor nodes and aggregate encrypted data. Then *BS* can decrypt aggregation result by its secret key. Albath and Madria [10] proposed an ECC-ElGamal based homomorphic encryption scheme to achieve confidentiality for in-network aggregation in wireless sensor networks. Even if the adversary compromises a node and obtains the public key, it cannot obtain the plaintext of intermediate aggregation results. Hence, public key-based homomorphic encryption schemes are resilient to node compromise attacks. However, the computational cost and communication overhead of public key encryption scheme are not quite tolerable for WSNs, especially when sensors are collecting diverse statistics (like temperature, humidity, and pressure).

Castelluccia et al. [6] first proposed an additively homomorphic encryption scheme which simply adds secret key k to environmental data x as ciphertext $c = x + k$. Each node has a unique secret key such that one node's corruption does not affect the data confidentiality of other nodes. Castelluccia et al. [11] improved their scheme in [11] by proposing a simple and provably secure encryption scheme that allows efficient additive aggregation of encrypted data. Each reporting node i encrypts plaintext data m_i as: $c_i = m_i + h(f_{ek_i}(r))$. The security of their scheme is based on the indistinguishability property of a pseudorandom function (PRF). However, ID-list of sensors has to be transferred and cannot be aggregated.

Feng et al. [9] tried to alleviate the ID-list problem and proposed a family of secret perturbation-based schemes that can protect sensor data confidentiality without disrupting the additive data aggregation result. BSP and FSP are two basic schemes which take nonredundant reporting approach and fully reporting approach, respectively. The ideal scheme O-ASP assumes that each sensor node knows the membership and topology of the whole network, and it knows whether each of these nodes has data satisfying each query. Then, *BS* computes aggregation and communication cost of two approaches for each cell before selecting one. To overcome the unrealistic assumption, D-ASP is proposed to enable nodes to make decisions based only on their locally available information, and interactions only take place within a cell or between neighboring cells. However, it is difficult for nodes to decide whether to report their ID with locally available information and it makes no difference when the number of reporting nodes is the same as nonreporting nodes. It also causes extra communication cost and network delay for waiting and deciding.

PRDA [15] pointed out that the transmission of sensor node IDs along with aggregated data packets increases the communication overhead of the network. Therefore, it keeps a table that consists of sensor node IDs and their corresponding small index numbers in each data aggregator. After the cluster forming, data aggregator generates the index table and sends it to *BS*. During data aggregation, instead of sending 2-byte sensor node IDs, data aggregators send corresponding index numbers. *BS* can find the ID of sensor nodes in the index table. However, index numbers are only used within clusters.

Although existing schemes tried to reduce the amount of IDs, they still suffer from related communication cost, and dropping ID or sending false ID will lead *BS* to compute false aggregation result.

Our work requires no ID to be forwarded and achieves a good trade-off between confidentiality and efficiency by adopting perturbation. With this improvement, we manage to simultaneously preserve data confidentiality and significantly reduce overall communication overhead, avoiding high energy consumption in aggregation phase.

3. System Model

3.1. Network Assumption. We assume a multilevel sensor network tree that consists of N (less than 1000) sensor nodes and certain amount of relay nodes. Sensor nodes are deployed in areas of interest, and they can sense and aggregate data. Both tree and cluster topologies can be applied in aggregation structure. In this paper, we use aggregation tree to illustrate our protocol. Aggregation tree could be formed as in TAG [4]. Relay nodes just forward messages, and they consist of a long path from targeted areas to *BS*. The powerful *BS* with transmission range covering the whole network is capable of broadcasting messages to all nodes directly. Each sensor node has a unique ID picked from the set $\{0, 1, \ldots, N-1\}$. After the aggregation tree is formed, each sensor node monitors its surrounded environment to generate environmental data which is an integer ranging from $[v_{\min}, v_{\max}]$. Environmental data (e.g., temperature) can be

converted to integers if necessary. Each reporting node and aggregator sends their messages up the aggregation tree. The message has the following format:

$$\langle c, hax \rangle , \quad (1)$$

where c is the number of reporting nodes in network and hax is the sum of environmental data and perturbation.

3.2. Design Goals. When designing confidentiality protection schemes, we aim to achieve the following goals.

(1) Data accuracy: *BS* can correctly retrieve the sum of environmental data with an overwhelming probability.

(2) Data confidentiality: the aggregation result should only be known by *BS* and PEC2P is CPA secure.

(3) Efficiency: the protocol should help to reduce communication overhead and prolong the network lifetime.

Definition 1 (Chosen Plaintext Attack). In this attack, the adversary has the ability to obtain the encryption of plaintexts of its choice. It then attempts to determine the plaintext that was encrypted in some other plaintext [17].

Definition 2 (Negligible Function). A function F is negligible if for every polynomial $p(\cdot)$, there exists an N such that for all integers $n > N$, it holds that $F(n) < 1/p(n)$. An equivalent formulation of the above is to require that for all constants c, there exists an N such that for all $n > N$, it holds that $F(n) < n^{-c}$.

We define an experiment for any private-key encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$, any adversary A, and any value n of security parameter.

The CPA Indistinguishability Experiment $PriK^{\text{CPA}}_{A,\Pi}(n)$.

(1) A random key k is generated by running $\text{Gen}(n)$.

(2) The adversary A is given input 1^n and oracle access to $\text{Enc}_k(\cdot)$, and outputs a pair of messages m_0, m_1 of the same length.

(3) A random bit $b \leftarrow \{0, 1\}$ is chosen, and then a ciphertext $c \leftarrow \text{Enc}_k(m_b)$ is computed and given to A. We call c the challenge ciphertext.

(4) A continues to have oracle access to $Enc_k(\cdot)$, and outputs a bit b'.

(5) The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. (A succeeds if $PriK^{\text{CPA}}_{A,\Pi}(n) = 1$).

Definition 3 (CPA secure). A private-key encryption scheme $\Pi = (\text{Gen}; \text{Enc}; \text{Dec})$ has indistinguishable encryptions under a chosen-plaintext attack (or is CPA secure) if for all probabilistic polynomial-time adversaries A there exists a negligible function negl such that:

$$Pr\left[PriK^{\text{CPA}}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + \text{negl}(n) . \quad (2)$$

3.3. aAttacker Model. We assume the existence of a global probabilistic polynomial time (PPT) adversary, which can choose to compromise a small subset of nodes and obtain all secrets of these nodes. With oracle access, it can also obtain the ciphertext for any chosen plaintext from any of the uncompromised nodes. Once the adversary compromises a sensor node, it will obtain its secret key and may modify, forge or discard messages, or simply transmit false aggregation results.

In this paper, we do not consider stealthy attacks [18] where the attacker's goal is to make the *BS* accept false aggregation results while not being detected. Also, we do not consider the denial-of-service (DoS) attack in various protocol layers [19, 20] where the adversary prevents the querier from getting any aggregation result at all. However, if a node does not respond to queries, it is clear that something is wrong, and solutions can be implemented to remedy this situation. Sybil/node replication attacks [21] or "wormhole" formation [22, 23] are beyond the scope of this paper.

4. PEC2P

The proposed scheme PEC2P mainly consists of bootstrapping phase, data aggregation phase, and result retrieving phase.

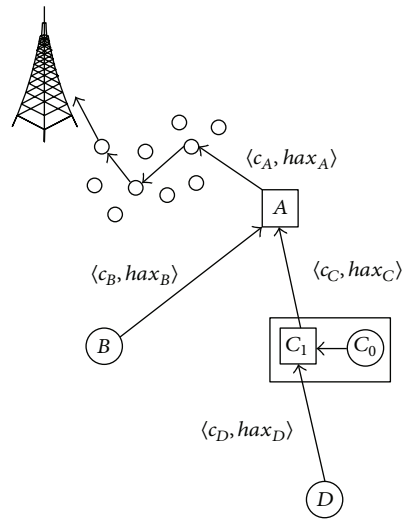
4.1. Bootstrapping Phase. In bootstrapping phase, modulus $M = 2^l$ is stored in all nodes, and so is a collision-resistant cryptographic hash function $H : \{0, 1\}^* \leftarrow \{0, 1\}^l$ and a PRF $f : \{0, 1\}^l \leftarrow \{0, 1\}^l$.

We further assume that *BS* first runs Algorithm 1 such that a unique initialization vector IV_i is generated, and secret key $k_i = IV_i$ is stored in *BS*'s local record and node i.

4.2. Data Aggregation Phase. Each sensor node in targeted area may behave as a sensing node, an aggregator, or combined. To simplify the discussion, we assume that each node can perform one role of sensing or aggregating without the loss of generality. Any node with combined role can be logically split into a sensing node and an aggregating node. As shown in Figure 2, aggregator C both senses data and aggregates data from downstream. It is divided into sensing node C_0 and aggregating node C_1. After the transformation, only leaf nodes sense environmental data.

In aggregation phase, when a targeted event happens or *BS* disseminates a query, each leaf sensor node i with environmental data x_i runs Algorithm 2 to compute individual aggregation result $\langle c_i, hax_i \rangle$. First, i inputs environmental data x_i, then sets $c_i = 1$ and $hax_i = x_i + H(k_i)$ since it has no children nodes. Second, i forwards the result to its parent node for data aggregation. Finally, i updates its secret key $k_i = f(k_i)$. Other leaf senor nodes remain hibernated.

During each aggregation, upon receiving a message from one of its children nodes for the first time, each aggregator i starts a timer t and collects other messages before t fires. Then, it runs Algorithm 3 to compute partial aggregation result $\langle c_i, hax_i \rangle$. First, i computes partial count $c_i = \sum_{j \in S_i} c_j$ and partial perturbed data $hax_i = \sum_{j \in S_i} hax_j \mod M$. Then



$x_D = 15$
$c_D = 1$
$hax_D = x_D + H_D \bmod M$
$= F9\ 37\ C3\ 8D$

$x_C = 10$
$c_C = 1 + C_D$
$hax_C = x_C + H_C + hax_D \bmod M$
$= 26\ F7\ 08\ 3D$

$x_B = 21, c_B = 1$
$hax_B = x_B + H_B \bmod M$
$= A4\ 6F\ E0\ D9$

$c_A = c_B + c_C = 3$
$hax_A = hax_B + hax_C \bmod M$
$= CB\ 66\ 19\ 16$

Figure 2: An example of data aggregation phase.

begin
randomly pick master key $SK \in \{0,1\}^l$;
for $i \leftarrow 0$ to $N-1$ **do**
$IV_i \leftarrow f_{SK}(i)$;
store IV_i in BS;
store $k_i = IV_i$ in BS and node i;
end

Algorithm 1: Bootstrapping algorithm.

begin
Input: environmental data x_i;
$c_i \leftarrow 1$;
$H_i \leftarrow H(k_i)$;
$hax_i \leftarrow (x_i + H_i) \bmod M$;
$k_i \leftarrow f(k_i)$;
return $\langle c_i, hax_i \rangle$;
end

Algorithm 2: Perturbation algorithm.

begin
$c_i \leftarrow \sum_{j \in S_i} c_j$;
$hax_i \leftarrow \sum_{j \in S_i} hax_j \bmod M$;
return $\langle c_i, hax_i \rangle$;
end

Algorithm 3: Aggregation algorithm.

i forwards the result to its parent node. Aggregators receiving no messages from downstream just remain hibernated. Note that we count number to trace the contributing nodes in BS; hence, synchronization among sensors is not needed.

Definition 4. S_i is a set of reporting node's ID, and these nodes are node i's children nodes.

To show how our scheme works, we take Figure 2 as an example. Node B and D are leaf sensor nodes with their own environmental data x_B and x_D. Node C is divided into C_0 and C_1 such that C_0 runs Algorithm 2 and node C_1 runs Algorithm 3 respectively. Aggregator A just forwards messages after aggregating data received from B and C. BS obtains the final aggregation result: $C_{BS} = 3$ and $HAX_{BS} = 0xCB66E916$.

4.3. Result Retrieving Phase. In result retrieving phase, after receiving final aggregation result $\langle C_{BS}, HAX_{BS} \rangle$, BS runs Algorithm 7 to retrieve ID list and actual aggregation result. First, BS orderly selects a list IDL of C_{BS} nodes and corresponding shared keys k_j from the N nodes, and BS computes

$$Agg = \left(HAX_{BS} - \sum_{i \in IDL} H(k_i) \right) \bmod M \tag{3}$$

if $Agg \in [C_{BS} * v_{\min}, C_{BS} * v_{\max}]$, and then BS will admit that Agg is the actual aggregation result $\sum_{i \in IDL} x_i$ and update secret keys for the found C_{BS} nodes. If not, BS will continue searching.

To improve searching efficiency for BS, we can first divide the network into clusters of trees each containing part of N nodes. Further analysis is in Section 5.3.

/* Define a private-key encryption scheme for messages of length L and key of length n as follows: */

(i) **Gen:** on input 1^n, choose $k \leftarrow \{0,1\}^n$ uniformly at random and output it as the key.

(ii) **Enc:** on input a key $k \leftarrow \{0,1\}^n$ and a message $m \leftarrow \{0,1\}^L$, output the ciphertext:

$$\langle c = 1,\ s = H(k)\rangle$$

(iii) **Dec:** on input a key $k \leftarrow \{0,1\}^n$ and a ciphertext $\langle c, s\rangle$, search for matching set S and output the plaintext:

$$m = s - \sum_{i\in S} H(k_i)$$

(iv) **Addition of Ciphertext:** given two ciphertext $\langle c_i, s_i\rangle$ and $\langle c_j, s_j\rangle$, output $\langle c_l, s_l\rangle$ as aggregation ciphertext:

$$c_l = c_i + c_j$$
$$s_l = \left(s_i + s_j\right) \bmod M$$

Algorithm 4: Construction Π^*.

5. Analysis and Experiments

5.1. Accuracy Analysis

Theorem 5. *PEC2P has a probability of at least*

$$1 - \frac{C_{BS} * \left(v_{\max} - v_{\min}\right)}{M - 1 - C_{BS} * v_{\min}} \tag{4}$$

in finding the correct combination in result the retrieving phase, given the environmental data in the range $[v_{\min}, v_{\max}]$, *the hash value in the range* $[0, 2^l - 1]$, *and modulus* M *is* 2^l.

Proof. We assume that $\{t \leftarrow \{0,1\}^\lambda : H(t)\}$ is the uniform distribution over $\{0,1\}^\lambda$, and then $H(x)$ is independent of $H(y)$ $(x \neq y)$. The probability of $H(x) = H(y)$ is $1/2^\lambda$. When $c = N$: $|T_1| = |T_2| = c, T_1 \neq T_2$, then we believe the probability of $\sum_{i\in T_1} \mathrm{Hash}(k_i) = \sum_{j\in T_2} \mathrm{Hash}(k_j)$ is $N/2^\lambda$.

Thus, adding C_{BS} environmental data together will result in a number in the range $[C_{BS} * v_{\min}, C_{BS} * v_{\max}]$, and the aggregation result is in the range $[C_{BS} * v_{\min}, C_{BS} * v_{\max} + M - 1] = [C_{BS} * v_{\min}, M - 1]$. If the result is valid, it has to belong to range $[C_{BS} * v_{\min}, C_{BS} * v_{\max}]$. Then, the probability that *BS* accepts a false aggregation result is at most

$$\frac{C_{BS} * \left(v_{\max} - v_{\min}\right)}{M - 1 - C_{BS} * v_{\min}}. \tag{5}$$

Hence, (4) holds. □

If we have 1024 nodes in the network and the data sensed from the environment is in the range $[0, 2^{32} - 1]$, we use SHA-1 as our hash function, and the output is in the range $[0, 2^{160} - 1]$. We can calculate the probability that *BS* accepts a false aggregation result is 2^{-118} which can be ignored.

We have implemented PEC2P using simple WSN experimental system to sense temperature in lab. Characteristics of SimpleWSN node is shown in Table 1.

Table 1: Characteristics of simple WSN node.

CPU 8-bit	8 MHz
Storage	10 Kbytes RAM 48 Kbytes FLASH
Communication	2.4 GHz
Bandwidth	250 Kbps
Operating system	TinyOS

Results are shown in Table 2. *BS* has ID $'01'$, and sensor node's ID $\in \{'01', '02', '03', '04', '05'\}$. Column 1 displays the number of participating nodes C from aggregation result $\langle C, HAX\rangle$. Column 2 displays the perturbed data HAX from $\langle C, HAX\rangle$. Column 3 displays the sum of hash value computed by *BS*. Column 4 displays the sum of environmental data after *BS* searching and subtracting the sum of hash value from HAX. The IDs of found nodes are shown in column 5. The temperature sensed is hexadecimal integer. We use Temperature (°C) $= ((t/4096) * 1.5 - 0.986)/(0.00355)$, provided by the SimpleWSN experimental platforms, to transform environment data to floating-point number which represent the Celsius degree. The average temperature is about 30 degrees Celsius in our experiment. The results justified the accuracy of PEC2P such that if we subtract data in column 3 from data in column 2, we will end up with data in column 4. The results verified that both the exact IDs and actual aggregation result are retrieved correctly.

5.2. Security Analysis.

We assume that each sensor node shares a unique key with *BS* and a common one-way hash function H is used. When an event happens, all nodes which are collecting environmental data will add the hash value computed on $f(k)$ to the environmental data x. Intuitively, since key k_i is only shared between node i and *BS*, other node j $(j \neq i)$ cannot successfully compute $H(k_i)$ with the probability ϵ that is not negligible. And it is also difficult for adversaries to compute the correct hash value of any given x. Hence, both privacy and confidentiality are achieved. We will prove this by security reduction. First, we construct an encryption scheme (Algorithm 4).

Table 2: Results of *BS* running selection algorithm after receiving aggregation results.

C	*HAX*				Hash sum				Raw Data				ID list				
03	50	63	46	27	50	63	23	3D	00	00	22	EA	00	03	00	05	06
03	49	77	C2	A4	49	77	9F	8F	00	00	23	15	00	03	04	00	06
02	56	A4	A8	A3	56	A4	91	38	00	00	17	6B	02	00	04	00	00
04	E1	1D	4A	91	E1	1D	1B	CA	00	00	2E	C7	02	03	04	00	06
01	2D	48	11	FF	2D	48	06	4B	00	00	0B	B4	00	00	04	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
04	BD	98	F6	19	BD	98	C7	8A	00	00	2E	8F	02	00	04	05	06
02	ED	48	FF	EE	ED	48	E8	78	00	00	17	76	00	03	04	00	00
03	A3	A6	20	EB	A3	A5	FD	F7	00	00	22	F4	02	00	04	05	00
01	5B	AD	89	B0	5B	AD	7D	EB	00	00	0B	C5	00	03	00	00	00
03	AE	AF	C4	01	AE	AF	A0	FF	00	00	23	02	02	03	00	05	00
01	EC	09	B6	05	EC	09	AA	6B	00	00	0B	9A	00	00	00	00	06
02	DB	23	9A	C9	DB	23	83	76	00	00	17	53	02	00	00	00	06
03	15	8B	95	E4	15	8B	72	ED	00	00	22	F7	02	00	04	05	00
02	16	E3	FB	2A	16	E3	E3	EC	00	00	17	3E	00	00	04	05	00
01	64	91	C9	E1	64	91	BE	29	00	00	0B	B8	02	00	00	00	00

Lemma 6. *Algorithm 4 is CPA secure if H is a pseudorandom function (PRF). One has*

$$Pr\left[PriK_{A,\Pi^*}^{CPA}(n) = 1\right] \leq \frac{1}{2} + negl(n). \tag{6}$$

Proof. If we replace the hash function H in Algorithm 4 with a truely random function F, we can have a new construction Π'. It is obvious that

$$Pr\left[PriK_{A,\Pi'}^{\mathrm{CPA}}(n) = 1\right] \leq \frac{1}{2} + \mathrm{negl}(n). \tag{7}$$

If H fulfills the requirement, then $\{t \leftarrow \{0,1\}^{\lambda} : H(t)\}$ is the uniform distribution over $\{0,1\}^{\lambda}$. Therefore, (6) holds. □

Theorem 7. *PEC2P is secure against CPA hash function if the following distributions are to be identical:*

$$\left\{t \leftarrow \{0,1\}^{\lambda} : H(t) + m_0\right\}, \left\{t \leftarrow \{0,1\}^{\lambda} : H(t) + m_1\right\}. \tag{8}$$

Proof. Proof for the nonhashed scheme. we assume that adversary A attacks (CPA) PEC2P with success probability $(1/2) + \epsilon(n)$. Now, we can construct a fast algorithm A' to "break" Construction Π^*, and A' tries to achieve its goal by running A as in Algorithm 5.

$$\begin{aligned}
&Pr_{A'}^{H}[\text{Success}] \\
&= \frac{1}{2}\left\{Pr\left[b'' = 0 \mid b = 0\right] + Pr\left[b'' = 1 \mid b = 1\right]\right\} \\
&= \frac{1}{2}\left\{\frac{1}{N}Pr\left[b'' = 0 \mid b = 0, b' = 0\right]\right. \\
&\qquad + \frac{N-1}{N}Pr\left[b'' = 0 \mid b = 0, b' = 1\right] \\
&\qquad + \frac{1}{N}Pr\left[b'' = 1 \mid b = 1, b' = 1\right] \\
&\qquad \left. + \frac{N-1}{N}Pr\left[b'' = 1 \mid b = 1, b' = 0\right]\right\} \\
&= \frac{1}{2}\left\{\frac{1}{N}Pr\left[PriK_{A,\mathrm{PEC2P}}^{\mathrm{CPA}}(n) = 1\right] + \frac{N-1}{N} * \frac{1}{2}\right. \\
&\qquad \left. + \frac{1}{N}Pr\left[PriK_{A,\mathrm{PEC2P}}^{\mathrm{CPA}}(n) = 1\right] + \frac{N-1}{N} * \frac{1}{2}\right\} \\
&= \frac{N-1}{N} * \frac{1}{2} + \frac{1}{N}Pr\left[PriK_{A,\mathrm{PEC2P}}^{\mathrm{CPA}}(n) = 1\right] \\
&= \frac{1}{2} + \frac{1}{N}\left(\frac{1}{2} + \epsilon(n)\right) = \frac{1}{2} + \frac{\epsilon(n)}{N}.
\end{aligned} \tag{9}$$

/* A' tries to break $Enc_k(x) = x + H(k)^*$ /
(1) A' initiates other $N - 1$ nodes and has access to N oracle $Enc(\cdot)$.
(2) A implements *PEC2P* l times and obtain the ciphertext of message x_i $(i = 1, 2, \ldots, l)$.
(3) A' forwards the queries to the network and return $H(f(k_i))$ to A.
(4) A outputs two messages m_0, m_1, sending them to A'.
(5) A random bit $b \leftarrow \{0, 1\}$ is chosen and A' makes an encryption query for m_b to $Enc_k(\cdot)$ and get back challenge ciphertext c_b $(b \in \{0, 1\})$.
(6) If c_b is from the node which holds secret key k, then A' returns c_b to A.
(7) A output a bit b' and returns it to A'.
(8) A' outputs $b'' = b'$.
(9) Else A' outputs $b'' = 0$ with the probability of 1/2 and outputs $b'' = 1$ with the probability of 1/2.
(10) Output 1 if $b'' = b$ and output 0 otherwise.

Algorithm 5: A'.

Input: $\langle IDList, HAX_{BS}\rangle$:
begin
  $Agg \leftarrow -1$;
  while $IDL \neq \perp$ **do**
    $hax_{BS} \leftarrow 0$; $j \leftarrow 0$;
    for $i \leftarrow 0$ to $N - 1$; **do**
      if $IDL[i] = 1$ **then**
        $hax_{BS} \leftarrow hax_{BS} + H(k_{ri})$;
    if $HAX_{BS} - hax_{BS} \in [C_{BS} * v_{\min},\ C_{BS} * v_{\max}]$
    then
      $Agg \leftarrow HAX_{BS} - hax_{BS}$;
    for $j \leftarrow 0$ to $C_{BS} - 1$ **do**
      $k_{\text{temp}[j]} \leftarrow f(k_{\text{temp}[j]})$;
    break;
    return Agg;
  return Agg;
end

Algorithm 6: Matching algorithm.

According to Lemma 6, we should have

$$\frac{1}{2} + \frac{\epsilon(n)}{N} \leq \frac{1}{2} + \text{negl}(n). \tag{10}$$

Therefore, $\epsilon(n) \leq \text{negl}(n)$.

Security of the Hashed Version. Only a few modifications to this security proof are needed in order to prove the security of the hashed variant. First, in Algorithm 5, all ciphertext are of now generated using the hashed values of k. Second, the security proof of the hashed scheme relies on the fact that $\{t \leftarrow \{0, 1\}^\lambda : H(t) + m_0\}$ and $\{t \leftarrow \{0, 1\}^\lambda : H(t) + m_1\}$ are identical distribution. If H fulfills the requirement, then $\{t \leftarrow \{0, 1\}^\lambda : H(t)\}$ is the uniform distribution over $\{0, 1\}^\lambda$. Consequently, the two distributions are identical. This thus concludes the proof that the hashed scheme is semantically secure. Thus, PEC2P is CPA secure. □

5.3. Efficiency Analysis. For a reporting leaf node, the computational cost only consists of one hash computation and one modular addition. For an aggregator, the computational cost consists of the sum operation of count and sum of perturbed data. If an aggregator has reporting data, it also has one hash computation.

Table 3: Number of bits sent per node for leaf node.

Protocol	Number of bits
O-ASP [All]	$\|h\| + 2 * \|\text{Per}\|$
O-ASP [Non]	$\|h\| + 2 * \|\text{Per}\| + \|\text{ID}\|$
Claude_09	$\|h\| + \|\text{Per}\|$
PEC2P	$h + \log_2 N + \|\text{Per}\|$

ID: node ID; h: header; Per: perturbed data; N: number of nodes in network.

We assume that there are N sensor nodes in reporting area and aggregation tree has a branching factor d of 3. Perturbed data Per = header + data + append. We choose the packet format used in TinyOS [16], and the packet header is 56 bits. Data is in the range of $[0, 127]$. Let count length, ID length, and append length be $\log_2 N$ bits. We consider two different scenarios: (1) only nodes at the lowest level may have data satisfying *BS*'s query and (2) nodes at each level may have data satisfying *BS*' query.

O-ASP [9] is designed based on an ideal and unrealistic assumption that each sensor node knows the membership and topology of the whole network and it knows whether each of these nodes has data satisfying each particular query. In each aggregation, a decision node (say *BS*) first compares the communication cost of [All-reporting] (A) and [Non-redundant-reporting] (N) for each cell and then decides which strategy will be chosen.

In Claude_09 [11], in the data aggregation phase, for scenario (1), each reporting node sends ($|\text{Per}|$) bits of message to its parent node, and nodes at second lowest level decide which group if IDs to send: the reporting nodes' IDs or the nonreporting nodes' IDs. For scenario (2), each reporting node will send ($|\text{ID}| + |\text{Per}|$) bits of message.

For PEC2P, in the data aggregation phase, for scenarios (1) and (2), each reporting node sends ($|\text{count}| + |\text{Per}|$) bits of message to its parent node, and the same length of message will also be sent from aggregators. No ID is transmitted in the aggregation tree.

We show the number of bits sent by leaf node in Table 3. Then, we calculate the average/maximum/minimum

Table 4: Theoretical analysis of total communication overhead.

(a)

	Claude_09
Average	$\sum_{i=\lceil \log_d N \rceil - 1}^{1} \left(\left(1 - (1-P)^{\sum_{j=1}^{\lceil \log_d N \rceil - i} d^j} \right) * \lvert H \rvert * d^i \right) + P * \lvert H \rvert * d^{\lceil \log_d N \rceil} + P' * \lvert \mathrm{ID} \rvert * d^{\lfloor \log_d N \rfloor} * \left(\lceil \log_d N \rceil - 1 \right)$
Minimum	$\lvert H \rvert * n + \lvert \mathrm{ID} \rvert * \left(\sum_{i=1}^{\lfloor \log_d n \rfloor} i * d^i + \lfloor \log_d n \rfloor * \left(n - d^{\lfloor \log_d n \rfloor} \right) \right)$
Maximum	
$C1$	$\lvert H \rvert * \left(\sum_{i=1}^{\lfloor \log_d n \rfloor} d^i + \left(n - d^{\lfloor \log_d n \rfloor} \right) * \left(\lfloor \log_d N \rfloor - \lfloor \log_d n \rfloor \right) + \lvert \mathrm{ID} \rvert * n * \lfloor \log_d N \rfloor \right)$
$C2$	$\lvert H \rvert * \left(N - d^{\lfloor \log_d N \rfloor} + n \right) + \lvert \mathrm{ID} \rvert * n * \lfloor \log_d N \rfloor$
$C3$	$\lvert H \rvert * N + \lvert \mathrm{ID} \rvert * \left(\sum_{j=\log_d \lceil \log_d N \rceil}^{i} j * d^j + (i-1) * n - S^i \right)$
	PEC2P
Average	$\sum_{i=\lceil \log_d N \rceil - 1}^{1} \left(1 - (1-P)^{\sum_{j=1}^{\lceil \log_d N \rceil - i} d^j} \right) * \lvert m \rvert * d^i + P * \lvert m \rvert * d^{\lceil \log_d N \rceil}$
Minimum	$\lvert m \rvert * n$
Maximum	
$C1$	$\lvert m \rvert * \left(\sum_{i=1}^{\lfloor \log_d n \rfloor} d^i + \left(n - d^{\lfloor \log_d n \rfloor} \right) * \left(\lceil \log_d N \rceil - \lfloor \log_d n \rfloor \right) \right)$
$C2$	$\lvert m \rvert * \left(N - d^{\lceil \log_d N \rceil} + n \right)$
$C3$	$\lvert m \rvert * N$

Note: only nodes at the lowest level may have data satisfying *BS*'s query.
$n = P * N$; d: degree; N: number of nodes in network; and $P' = P$ if $P \leq 0.5$ or $P' = 1 - P$.
ID: node ID; H = header + data + appendedBit; m = count + header + data + appendedBit.
C_1: $n < d^{\lfloor \log_d N \rfloor - 1}$; C_2: $d^{\lfloor \log_d N \rfloor - 1} < n < d^{\lfloor \log_d N \rfloor}$; C_3: $n > d^{\lfloor \log_d N \rfloor}$; $S_j = \sum_{\log_d \lceil N \rceil}^{j} d^j$, and $S^i < n < S^{i-1}$.

(b)

	Claude_09
Average	$\sum_{i=\lceil \log_d N \rceil - 1}^{1} \left(P * \lvert \mathrm{ID} \rvert * d^i * (i-1) + \left(1 - (1-P)^{\sum_{j=0}^{\lceil \log_d N \rceil - i} d^j} * \lvert H \rvert * d^{i-1} \right) \right) + P * d^{\lceil \log_d N \rceil} * \lvert \mathrm{ID} \rvert * \left(\lceil \log_d N \rceil - 1 \right) + P * \lvert H \rvert * d^{\lceil \log_d N \rceil}$
Minimum	$\lvert H \rvert * n + \lvert \mathrm{ID} \rvert * \left(\sum_{i=1}^{\lfloor \log_d n \rfloor} i * d^i + \left(\lfloor \log_d n \rfloor + 1 \right) * \left(n - d^{\lfloor \log_d n \rfloor} \right) \right)$
Maximum	
$C1$	$\lvert H \rvert * \left(\sum_{i=1}^{\lfloor \log_d n \rfloor} d^i + \left(n - d^{\lfloor \log_d n \rfloor} \right) * \left(\lfloor \log_d N \rfloor - \left(\lfloor \log_d n \rfloor \right) \right) + \lvert \mathrm{ID} \rvert * n * \lfloor \log_d N \rfloor \right)$
$C2$	$\lvert H \rvert * \left(N - d^{\lfloor \log_d N \rfloor} + n \right) + \lvert \mathrm{ID} \rvert * n * \lfloor \log_d N \rfloor$
$C3$	$\lvert H \rvert * N + \lvert \mathrm{ID} \rvert * \left(\sum_{j=\log_d \lceil \log_d N \rceil}^{i} j * d^j + (i-1) * n - S^i \right)$
	PEC2P
Average	$\sum_{i=\lceil \log_d N \rceil - 1}^{1} \left(1 - (1-P)^{\sum_{j=0}^{\lceil \log_d N \rceil - i} d^j) * \lvert m \rvert * d^{i-1}} + P * \lvert m \rvert * d^{\lceil \log_d N \rceil} \right)$
Minimum	$\lvert m \rvert * n$
Maximum	
$C1$	$\lvert m \rvert * \left(\sum_{i=1}^{\lfloor \log_d n \rfloor} d^i + \left(n - d^{\lfloor \log_d n \rfloor} \right) * \left(\lceil \log_d N \rceil - \lfloor \log_d n \rfloor \right) \right)$
$C2$	$\lvert m \rvert * \left(N - d^{\lceil \log_d N \rceil} + n \right)$
$C3$	$\lvert m \rvert * N$

Note: nodes at each level may have data satisfying *BS*'s query.
$n = P * N$; d: degree; N: number of nodes in network; and $P' = P$ if $P \leq 0.5$ or $P' = 1 - P$.
ID: node ID; H = header + data + appendedBit; m = count + header + data + appendedBit.
C_1: $n < d^{\lfloor \log_d N \rfloor - 1}$; C_2: $d^{\lfloor \log_d N \rfloor - 1} < n < d^{\lfloor \log_d N \rfloor}$; C_3: $n > d^{\lfloor \log_d N \rfloor}$; $S_j = \sum_{\log_d \lceil N \rceil}^{j} d^j$, and $S^i < n < S^{i-1}$.

Table 5: Number of bits sent per node for each level with Claude_09 scheme.

Level	Number node	A (100%)	A (90%)	A (70%)	AV (100%)	AV (90%)	AV (70%)	HBH-A	HBH-AV	No-Agg
1	3	75	949.8	2699.4	100	974.8	2724.4	73	97	68859
2	9	75	366.6	949.8	100	391.6	974.8	72	94	22932
3	27	75	172.2	366.6	100	197.2	391.6	70	91	7623
4	81	75	107.4	172.2	100	132.4	197.2	68	87	2520
5	243	75	85.8	107.4	100	110.8	132.4	67	84	819
6	729	75	78.5	83.8	100	103.5	108.1	65	81	252
7	2187	75	67.5	52.5	100	90	70	63	63	63

Note: only the nodes in the lowest level may have data satisfying *BS*'s query.

Table 6: Number of bits sent per node for each level with PEC2P scheme.

Level	Number Node	A (100%)	A (90%)	A (70%)	AV (100%)	AV (90%)	AV (70%)	HBH-A	HBH-AV	No-Agg
1	3	87	87	87	112	112	112	73	97	68859
2	9	87	87	87	112	112	112	72	94	22932
3	27	87	87	87	112	112	112	70	91	7623
4	81	87	87	87	112	112	112	68	87	2520
5	243	87	87	87	112	112	112	67	84	819
6	729	87	86.9	84.7	112	111.9	109	65	81	252
7	2187	87	78.3	60.9	112	100.8	78.4	63	63	63

Note: only nodes in the lowest level may have data satisfying *BS*'s query.

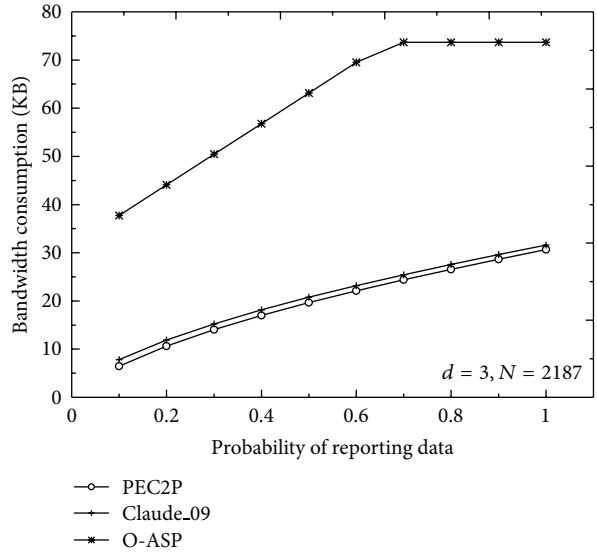


Figure 3: Communication overhead with different probability of reporting data when only nodes in the lowest level may have data satisfying *BS*' query.

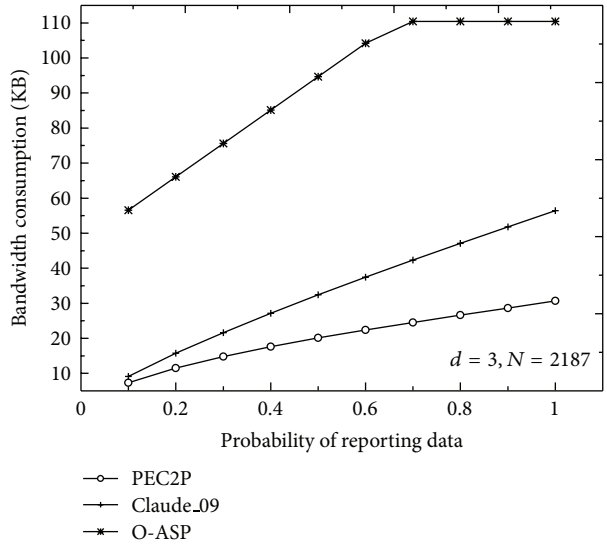


Figure 4: Communication overhead with different probability of reporting data when nodes at each level may have data satisfying *BS*' query.

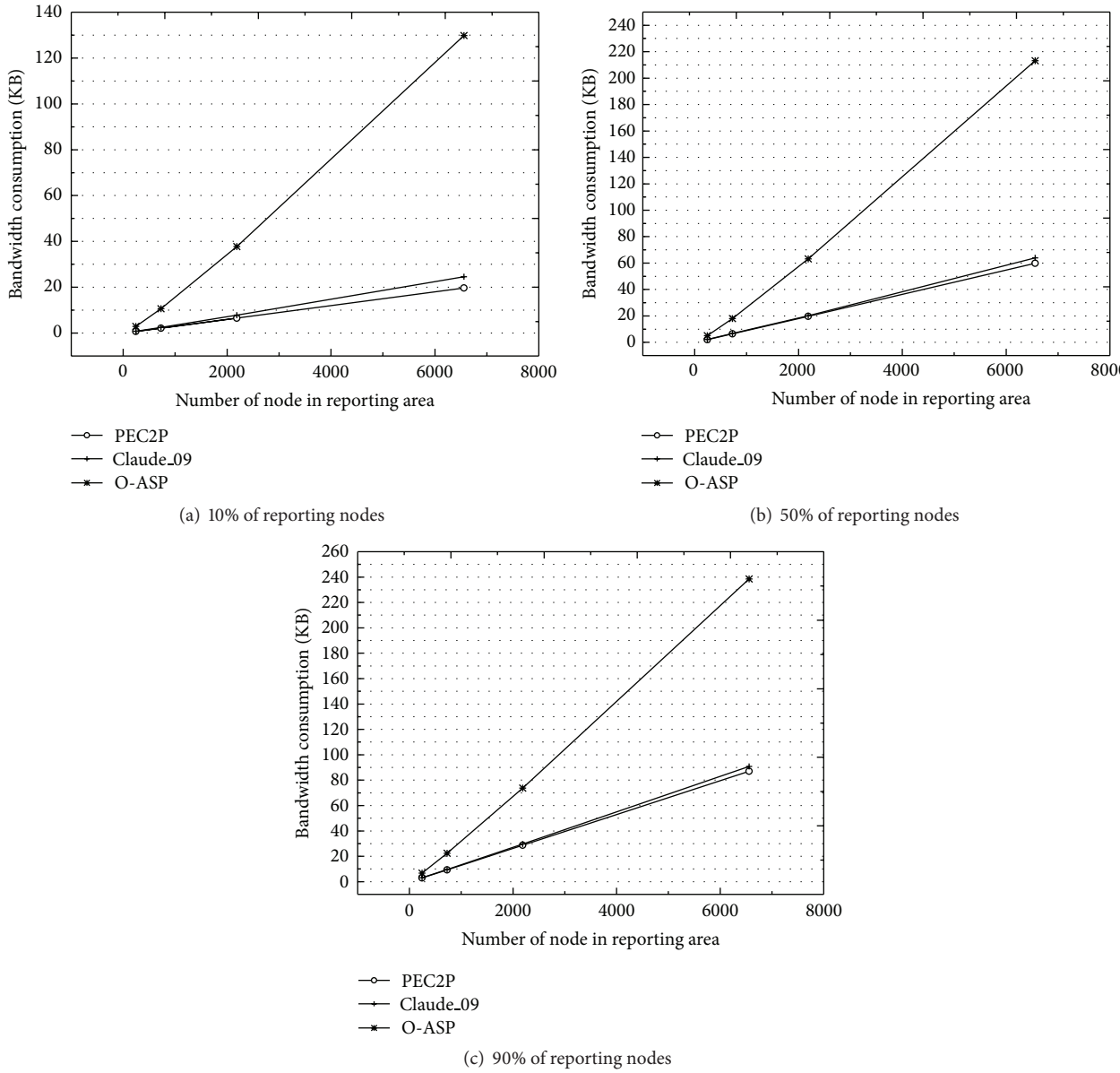


Figure 5: Bandwidth consumption in data aggregation phase when only nodes in the lowest level may have data satisfying *BS*' query.

communication overhead CO in aggregation phase for Claude_09 and PEC2P in Table 4. In minimum case, reporting nodes are located in the high levels of aggregation tree, and we can find them through breadth-first search. In maximum case, reporting nodes should be located from the lowest level to higher levels. Tables 5 and 6 list the number of bits sent per node for each level with Claude_09 and PEC2P.

Figures 3 and 4 show the trend of communication overhead in two different scenarios.

We assume that only the nodes in the lowest level have a probability of $P(= 0.1, 0.5, 0.9)$ to sense environmental data. Results are shown in Figures 5(a), 5(b), and 5(c).

We further assume that all nodes in aggregation tree has a probability of $P(= 0.1, 0.5, 0.9)$ to sense environmental data. Results are shown in Figures 6(a), 6(b), and 6(c).

Results show that, compared with existing protocols, PEC2P can greatly reduce communication overhead in aggregation phase. We notice that the major communication overhead is caused by transferring the hash value which was computed by SHA-1 in the comparison. Performance can be further optimized by choosing other hash functions with shorter output in case of lower security level requirement.

Result Retrieving Algorithm Test. We used a computer with a Pentium(R) D CPU of 3.40 GHZ and 2.00 GB memory to test Algorithm 7. Since sensor nodes are relatively uniformly distributed and their communication range is from 50 meters to 100 meters, a local event will be detected by a small group of sensor nodes. Therefore, we choose to use a small N. Results show that choosing 5 nodes from 10 nodes only needs 8 milliseconds and choosing 10 nodes from 20 nodes only needs

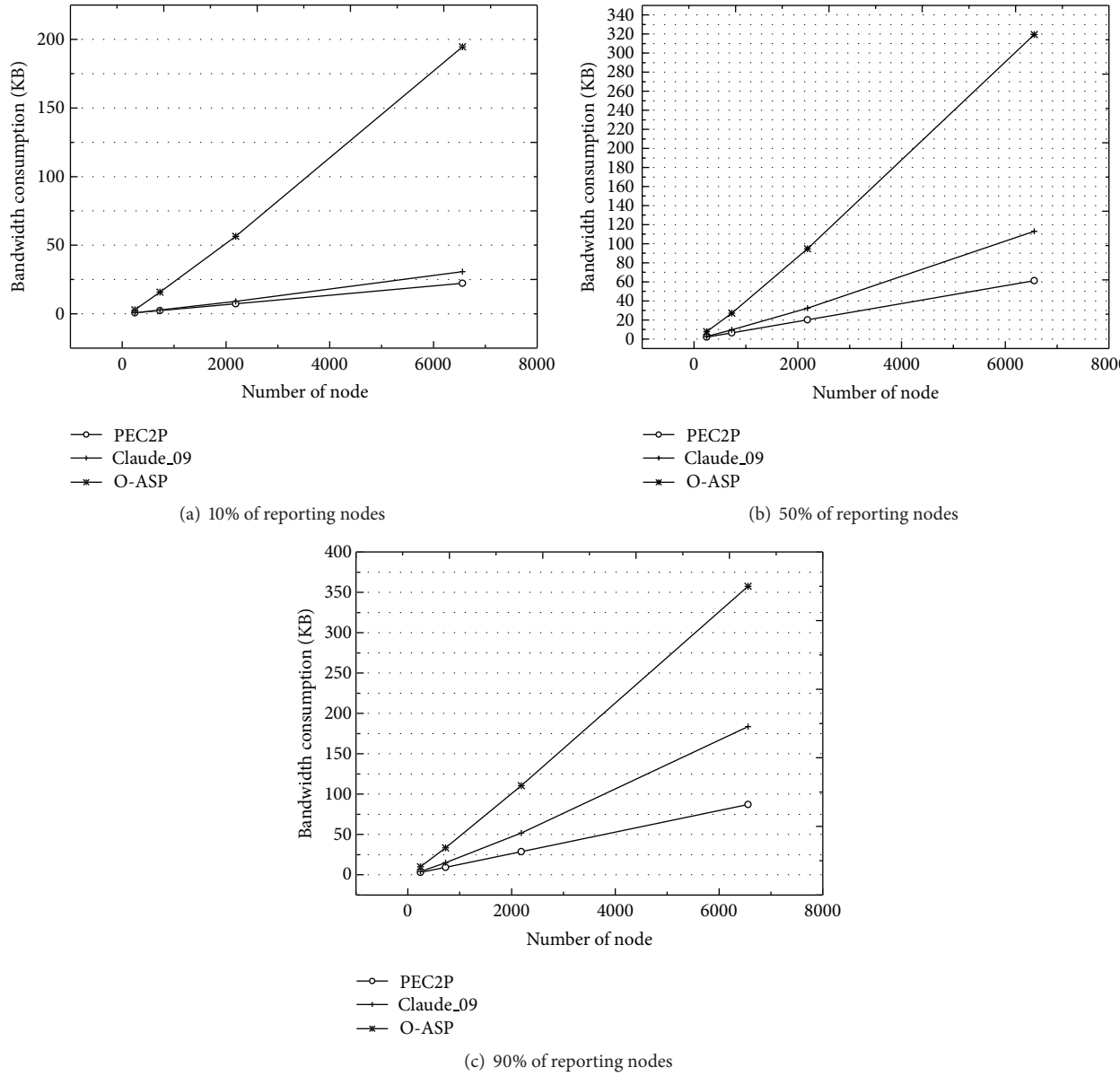


Figure 6: bandwidth consumption in data aggregation phase when nodes at each level may have data satisfying *BS*' query.

approximately 2 seconds. In WSNs, the capability of *BS* is more powerful than our experimental computer; thus, the searching time will be shorter in real applications. To make the search efficiently, we can first divide the network into clusters of trees.

6. Conclusion

Confidentiality protection and energy efficiency are two conflict, but equally crucial requirements in WSNs. To achieve a trade-off between these two goals simultaneously, remains a challenge. We propose PEC2P to protect data confidentiality which also achieves energy efficiency. Specifically, we need no ID list and use one-way hash function as perturbation added to the environmental data. Since *BS* usually has powerful computation capacities, we utilize *BS* to the fullest and let it compute which nodes have actually contributed to the aggregation process after receiving the final perturbed aggregation result. Consequently, we manage to preserve data confidentiality, avoid high energy consumption, and obtain lower overall communication overhead. Analysis and experiments have also been conducted to evaluate the proposed protocol. The results show that our protocol provides confidentiality protection for both raw and aggregated data with an overhead lower than that of the existing related protocols. PEC2P can be adopted to tree/cluster-based aggregation and any protocol using ID-list transmission. We focus on collecting the number of contributing nodes and its perturbed data, instead of how the information is gathered. For uniformity,

```
begin
  Agg ← −1; i ← 0; c ← 0;
  for i ← 0 to C − 1 do
    IDL[i] ← 1;
  for i: C to N − 1 do
  IDL[i] ← 0;
  Agg ← Matching(IDL, HAX_BS);
  if Agg ≠ − 1 then
    return ⟨IDL, Agg⟩;
  /*search C_N^C − 1 times*/
  for order ← 1 to C_N^C − 1 do
    for i ← 0 to N − 1 do
      if IDL[i] = 1 then
        c ← c + 1;
        /*find the last '1' in IDL[ ]*/
        if c = C and i ≤ N − 1 then
          IDL[i + 1] ← 1; IDL[i] ← 0;
          Agg ←Matching(IDL, HAX_BS);
          if Agg ≠ − 1 then
            return ⟨IDL, Agg⟩;
          i ← N; c ← 0;
        /*The last '1' is in the last position, then
        move the last continuous '1's to the first
        found '1' before them*/
        if c = C and i = N − 1 then
          /how many '1's should be moved*/
          group ← 0;
          /*from behind*/
          for j ← N − 1 to 0 do
            if IDL[j] = 1 then
              group ← group + 1;
            /*found the empty position and
            move the newly found '1' and the
            continuous '1's*/
            else
              /*newly found '1'*/
              m ← 0;
              /continuous '1's' new location*/
              NewLocation ← 0;
              for m ← j − 1 to 0 do
                if IDL[m] = 1 then
                  IDL[m] ← 0;
                  IDL[m + 1] ← 1;
                  NewLocation ← m + 2;
                  /*searching ends*/
                  m ← −1;
            /*move the continuous '1's*/
            if NewLocation < N then
              for p ← N − 1 to N − group
              do
                IDL[p] ← 0;
              for q ← NewLocation to
              group + NewLocation − 1 do
                IDL[q] ← 1;
            j ← −1;
        Agg ←Matching(IDL, HAX_BS);
        if Agg ≠ − 1 then
          return ⟨IDL, Agg⟩;
          i ← N; c ← 0;
end
```

Algorithm 7: Result retrieving algorithm.

we use tree topology in our paper. We also did cluster-based comparison with existing protocols, and the results show no significant difference.

Appendix

For more details, see Algorithms 6 and 7.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (nos. 61272512, 61003262, and 61100172), Program for New Century Excellent Talents in University (NCET-12-0047), and Beijing Natural Science Foundation (no. 4121001).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] M. Li and Y. Liu, "Iso-Map: energy-efficient contour mapping in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 699–710, 2010.

[4] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation ACM SIGOPS Operating Systems Review (OSDI '02)*, vol. 36, pp. 131–146, 2002.

[5] K. Akkaya, M. Demirbas, and R. S. Aygun, "The impact of data aggregation on the performance of wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 2, pp. 171–193, 2008.

[6] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems—Networking and Services (MobiQuitous '05)*, pp. 109–117, July 2005.

[7] J. Girao, D. Westhoff, and M. Schneider, "CDA: concealed data aggregation for reverse multicast traffic in wireless sensor networks," in *Proceedings of IEEE International Conference on Communications (ICC '05)*, pp. 3044–3049, May 2005.

[8] E. Mykletun, J. Girao, and D. Westhoff, "Public key based cryptoschemes for data concealment in wireless sensor networks," in *Proceedings of the 41st IEEE International Conference on Communications (ICC '06)*, pp. 2288–2295, July 2006.

[9] T. Feng, C. Wang, W. Zhang, and L. Ruan, "Condentiality protection for distributed sensor data aggregation," in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM '08)*, pp. 68–76, 2008.

[10] J. Albath and S. Madria, "Secure hierarchical data aggregation in wireless sensor networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '09)*, pp. 1–6, April 2009.

[11] C. Castelluccia, A. C. F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–36, 2009.

[12] S. Ozdemir and Y. Xiao, "Secure data aggregation in wireless sensor networks: a comprehensive overview," *Computer Networks*, vol. 53, no. 12, pp. 2022–2037, 2009.

[13] L. Hu and D. Evans, "Secure aggregation for wireless networks," in *Proceedings of the 3rd IEEE Symposium on Applications and the Internet Workshops (SAINT '03)*, p. 384, IEEE Computer Society, Washington, DC, USA.

[14] A. Mahimkar and T. S. Rappaport, "SecureDAV: a secure data aggregation and verification protocol for sensor networks," in *Proceedings of the 47th IEEE Global Telecommunications Conference (GLOBECOM'04)*, pp. 2175–2179, December 2004.

[15] S. Ozdemir and Y. Xiao, "Polynomial regression based secure data aggregation for wireless sensor networks," in *Proceedings of the 54th IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–5, 2011.

[16] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, November 2004.

[17] J. Katz and A. Y. Lindell, *Modern Crytography*, Chapman & Hall, 2008.

[18] B. Przydatek, D. Song, and A. Perrig, "SIA: secure information aggregation in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 255–265, November 2003.

[19] J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter, "Detection of denial-of-message attacks on sensor network broadcasts," in *Proceedings of the 25th IEEE Symposium on Security and Privacy (SP '05)*, pp. 64–78, May 2005.

[20] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[21] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: analysis and defenses," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 259–268, April 2004.

[22] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proceedings of the 25th IEEE Symposium on Security and Privacy (SP '05)*, pp. 49–63, May 2005.

[23] S. Brands and D. Chaum, "Distance-bounding protocols," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT '93)*, vol. 839 of *Lecture Notes in Computer Science*, pp. 344–359, Springer, 1994.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 615906, 9 pages
http://dx.doi.org/10.1155/2013/615906

Research Article

A Distributed Pseudonym Management Scheme in VANETs

Xiaoling Zhu,[1] Yang Lu,[1,2] Benhong Zhang,[1] and Zhengfeng Hou[1]

[1] *School of Computer and Information, Hefei University of Technology, Hefei 230009, China*
[2] *The Anhui Provincial Key Laboratory of Mine IoT and Mine Safety Supervisory Control, Hefei 230088, China*

Correspondence should be addressed to Xiaoling Zhu; zhuxl@hfut.edu.cn

Received 22 December 2012; Accepted 3 April 2013

Academic Editor: Xu Yongjun

Security and privacy have been important issues in VANETs. Anonymity is an effective way to achieve privacy protection, and it sometimes requires to be disclosed for determining traffic liability. In most pseudonym schemes, an authority is aware of a vehicle's secret, and its compromise will result in the leakage of a large amount of privacy information. So, we propose a distributed traceable pseudonym management scheme in VANETs. In the scheme, a blind signature method is adopted to achieve strict separation of issuance and tracking. The distributed tracking protocol is proposed to enhance the robustness for tracking, which is based on the improved scheme for shared generation of RSA keys. An efficient pseudonymous authentication mechanism is proposed to reduce the communication overhead. Compared with other related proposals, our scheme is unforgeability, especially against authority forge attacks, and has better robustness. Moreover, the performance analysis shows that it is efficient in VANETs.

1. Introduction

Vehicle-to-vehicle and vehicle-to-infrastructure communications improve vehicle's perception from the surrounding environment. Vehicular ad hoc networks (VANETs) will be used widely in collision avoidance, road-hazard notification, and coordinated driving systems [1]. Nonetheless, there are many security threats in VANETs [2–4]. For example, an attacker might tamper with messages to evade accident liability or forge information to meet specific needs. The attacker also might eavesdrop on broadcast messages, analyze data, and track a vehicle. So, security and privacy have been important issues in VANETs.

A number of studies have been made on the issues of security and privacy preservation. Raya and Hubaux [5, 6] pointed out that anonymity is conditional for liability purposes and that authority can disclose the pseudonym. In [5, 6], a security protocol was introduced. Although this protocol can effectively meet the conditional privacy requirement, it is far from efficient and can hardly become a scalable and reliable approach, because the authority has to keep all the anonymous certificates for each vehicle. Lin et al. [7] proposed a security and privacy preserving protocol. With group signature, security, privacy, and traceability can be achieved without inducing the overhead of managing a huge number of stored certificates at the authorities' sides. Calandriello et al. [8, 9] proposed on-the-fly pseudonym generation and self-certification, which alleviates the overhead of managing certificates. Group signature method is adopted to ensure that legitimate nodes can generate their pseudonyms anonymously. Lu et al. [10] presented a conditional privacy preservation protocol, which improves efficiency in terms of the minimized anonymous keys storage at each vehicle. Performance evaluation shows that the protocol can achieve much better efficiency than Raya and Hubaux's [5, 6] and Lin et al.'s [7] when vehicles are revoked. Zhang et al. [11] proposed a scalable robust authentication protocol. In [11], some roadside units (RSUs) serve as the issuer of vehicles' private key, and a signcryption method is employed to distribute the keys securely. Hao et al. [12] proposed a distributed key management framework, which has advantages in the revocation of malicious vehicles and system maintenance. An efficient cooperative message authentication protocol is developed to reduce the computation and communication overhead in the group signature.

The above reported schemes [7–12] are based on group signature. In Boneh et al.'s group signature scheme [13], each user's private key is generated by the private-key issuer, which

is a hidden security threat. In [7–9], each vehicle's group private key is computed by a member manager. In [10], a trusted authority is required; the authority generates valid private keys for on-board unit and RSU. In [11], RSU generates and sends the group private key to the vehicle. In [12], some measures are adopted to prevent RSU from misbehaving, but authorities cannot decide which is the malicious, RSU or the vehicle or both, when they find a mismatch. Therefore, these schemes [7–12] suffered from private key revealing attacks, in which the private-key issuer knows each user's private key.

Schaub et al. [14] adopted blind signature technology to achieve the separation of issuing and tracking. But the disadvantages of the scheme are that V-token ($V_i = E_{\mathrm{PK}_{\mathrm{RA}}}(id \parallel r_i)$) is produced by a vehicle alone; thus, the vehicle might forge other vehicle's V-token. Moreover, if CA is not credible, it can generate V-token by itself and sign it. CA can obtain the pseudonym certificate from a pseudonym provider and impersonate any vehicle.

To solve the above problems, this paper presents a distributed pseudonym management scheme in secure VANETs. The main contributions of the scheme are as follows. (1) Pseudonym is coproduced by the issuer and the vehicle. Either party attempting to deceive can be detected. It can resist authority forge attacks. (2) An efficient pseudonym authentication mechanism is proposed by finding the optimal number of messages with the pseudonym certificate, which not only reduces the communication overhead but also ensures the message authentication probability $P_{\mathrm{auth}} > 95\%$. (3) Distributed pseudonym tracking based on secret sharing method is presented. The initialization of the tracking protocol does not require a trusted center, thus avoiding any single point of failure. It offers better robustness.

The remainder of this paper is organized as follows. The pseudonym management model is given in Section 2. The pseudonym issuance protocol, the pseudonymous authentication protocol, and the distributed tracking protocol are presented in Sections 3, 4, and 5, respectively. Section 6 analyzes and compares the security and the performance of our scheme with other related schemes. Finally, the conclusion of this paper is given in Section 7.

2. Pseudonym Management Model

There are three types of entities: (1) a vehicle (V). Its identity is ID_V, corresponding to a long-term public key (N_V, e_V) and a long-term private key d_V. Its pseudonym is $\mathrm{ID}_{\mathrm{PV}}$, corresponding to a short-term public key $(N_{\mathrm{PV}}, e_{\mathrm{PV}})$ and a short-term private key d_{PV}. The vehicle contains a sensing input module, a wireless communication module, a central processing module, and a hardware security module (HSM) [15]. The HSM generates public and private keys, stores private keys, and provides digital signature service. (2) An authority: it is divided into certificate authority (CA), pseudonym certificate authority (PCA), and tracking authority (TA). PCA issues pseudonym certificates, but it does not know the pseudonym. Only TA knows the relation between the pseudonym and the identity. CA issues Cert_V, and Cert_V contains ID_V and (N_V, e_V). PCA issues $\mathrm{Cert}_{\mathrm{PV}}$, and $\mathrm{Cert}_{\mathrm{PV}}$ contains $\mathrm{ID}_{\mathrm{PV}}$ and $(N_{\mathrm{PV}}, e_{\mathrm{PV}})$. PCA's public and private keys are denoted as $(N_{\mathrm{PCA}}, e_{\mathrm{PCA}})$ and d_{PCA}, respectively. TA's public and private keys are denoted as (N, e) and d, respectively. There are some PCAs in the model, and V may apply for a pseudonym to the neighboring PCA. Only a few suspicious vehicles need to be disclosed, so TAs are rare. But if only one authority acts as TA, abuse can occur. Based on the secret sharing method, we extend one authority to k authorities forming $\mathrm{TAs} = \{\mathrm{TA}_1, \mathrm{TA}_2, \ldots, \mathrm{TA}_k\}$. TA_i $(1 \leqslant i \leqslant k)$ may be a law enforcement agency, a judge, or a privacy protection agency. TA_i's identity is ID_i. Assume that the vehicle is preloaded with the public keys of CA, PCA, and TAs during the vehicle's initialization. (3) Roadside unit (RSU): it communicates with vehicles and other devices on the internet.

The security and privacy requirements in the model are as follows.

Anonymity. For other entities (such as PCA, attackers) except TAs, it is computationally infeasible to disclose the identity from a pseudonym.

Traceability. If the members in TAs implement the protocol honestly, at least m members can disclose collaboratively the identity from a pseudonym.

Unforgeability. For any entity, it is computationally infeasible to forge a false signature or impersonate another entity.

Robustness. If any authority compromises, the implementation of pseudonym issuance, pseudonymous authentication, and tracking are not affected.

The process of pseudonym management is shown in Figure 1. (1) The vehicle V gets an identity certification from CA. (2) V applies for pseudonym certificates. (3) PCA issues some pseudonym certificates to V. (4) V communicates with other vehicles and RSU with the pseudonym certificates. (5) Once other vehicles find suspicious vehicles, they submit a tracking request to TAs. (6) TAs disclose the identity from the pseudonym.

The model consists of three protocols: a pseudonym issuance protocol, a pseudonymous authentication protocol, and a distributed tracking protocol. TAs' public key (N, e) is generated during the initialization stage and used in the pseudonym issuance protocol. Pseudonym certificates are generated in the issuance protocol and used in the pseudonymous authentication protocol. Once suspicious messages appear in pseudonymous authentication, the distributed tracking protocol is activated.

3. Pseudonym Issuance Protocol

Chaum [16] first proposed the concept of a blind signature, which allows users to get a message signature without leaking any contents. The pseudonym certificate issuance protocol adopts the blind signature method.

(1) V sends $\mathrm{ID}_V \parallel n \parallel (\mathrm{ID}_V \parallel n)^{d_v} \bmod N_V \parallel \mathrm{Cert}_V$ to PCA, where n is the number of pseudonyms.

(2) PCA verifies $(\mathrm{ID}_V \parallel n)^{d_v} \bmod N_V$, if passed, and sends $\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp \parallel ((\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}})^{e_V} \bmod N_V$ to V, where exp is the expiration date.

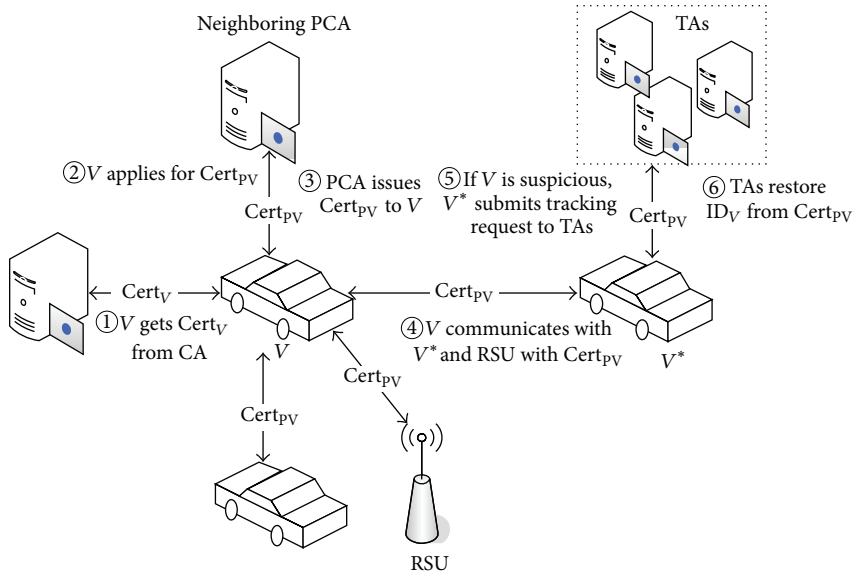


Figure 1: Pseudonym management framework.

(3) V extracts $((\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}})^{e_V} \bmod N_V$, decrypts it, and gets $(\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}}$. Then V verifies PCA's signature, if passed, picks two random integers r_i and x_i $(1 \leqslant i \leqslant 2n)$, and generates the pseudonyms as follows:

$$\mathrm{ID}_{\mathrm{PV}_i} = \left(r_i \parallel r_i^{d_v} \bmod N_V \parallel (\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}}\right)^{e} \bmod N, \quad (1)$$

V extends them to

$$b_i = \mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_{\mathrm{PV}_i} \parallel \exp \parallel e_{\mathrm{PV}_i} \parallel N_{\mathrm{PV}_i}. \quad (2)$$

V sends PCA the blind alternative commitments $c_i = (x_i^{e_{\mathrm{PCA}}}) \cdot \mathrm{SHA}(b_i) \bmod N_{\mathrm{PCA}}$ $(1 \leqslant i \leqslant 2n)$, where SHA is a message digest function.

(4) PCA randomly generates verification set I, $I = \{i \mid i \in [1, 2n]\}$, and $|I| = n$, and sends I to V.

(5) V shows $\{(r_i, r_i^{d_v}, x_i, e_{\mathrm{PV}_i}, N_{\mathrm{PV}_i}) \mid i \in I\}$ to PCA.

(6) PCA computes $\mathrm{ID}'_{\mathrm{PV}_i}$, b'_i and $c'_i = (x_i^{e_{\mathrm{PCA}}}) \cdot \mathrm{SHA}(b'_i) \bmod N_{\mathrm{PCA}}$ and checks $c'_i = c_i$. If passed, PCA sends the blind signatures $\{c_j^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}} \mid j \notin I\}$ of the remaining commitments to V.

(7) V removes the blind factors $c_j^{d_{\mathrm{PCA}}}/x_j = \mathrm{SHA}(b_j)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}} (j \notin I)$ and gets the pseudonym certificates as follows:

$$\mathrm{Cert}_{\mathrm{PV}_j} = b_j \parallel \mathrm{SHA}\left(b_j\right)^{d_{\mathrm{PCA}}}. \quad (3)$$

The prerequisite of implementing the protocol is that during the initialization stage, V has got TAs' public key (N, e) and PCA's public key $(N_{\mathrm{PCA}}, e_{\mathrm{PCA}})$. Before applying for a pseudonym, the short-time public and private keys of the vehicle have been already generated. (1) V sends a request signed with d_V to PCA to prove its identity ID_V. (2) PCA sends the signature secretly, not only to prove PCA's identity but also to coproduce V's pseudonym. Sending the signature secretly can prevent the signature from leaking. (3) V produces pseudonyms and blinds them. In (4), (5), and (6) PCA opens n commitments among $2n$ commitments to verify. If passed, PCA signs blindly the remaining n commitments. (7) V removes the blind factors and gets the pseudonym certificates.

In Schaub et al.'s issuance protocol [14], CA signs V-tokens for a vehicle. Pseudonym provider (PP) checks the validity of V-tokens; if valid, PP issues a pseudonym certificate. Four rounds of interaction are required. If CA wants to cheat, it will not interact with a vehicle. And it will obtain a fake pseudonym certificate and impersonate a vehicle, so will PP.

In our issuance protocol, PCA issues directly a pseudonym certificate to a vehicle, and three rounds of interaction are required. A pseudonym is coproduced by V and PCA. PCA cannot provide $r_i \parallel r_i^{d_v}$ to forge a pseudonym of V. V cannot provide $(\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_{V^*} \parallel \exp)^{d_{\mathrm{PCA}}}$ to forge the pseudonym of another vehicle V^*. Even if V eavesdrops on the communication between V^* and PCA, V cannot decrypt $((\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_{V^*} \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}})^{e_{V^*}}$. So, it fails to forge the pseudonym. Furthermore, the cut-choose method is also adopted in our protocol, like in Schaub et al.'s protocol, and prevents content spoofing to a certain extent.

As a result, our issuance protocol maintains good property of vehicular privacy in the presence of an authority and provides security against authority forge attack. Moreover, the communication overhead is reduced from four rounds to three rounds.

4. Pseudonymous Authentication Protocol

A complete authentication message consists of six fields: messageID ‖ payload ‖ timestamp ‖ signature ‖ certificate ‖ TTL. "MessageID" is the message number during the same pseudonym period. "Payload" contains collision data, location, direction, speed, and so on. "Timestamp" is to prevent from replaying attacks. "Signature" is the signature of the first three fields. The next field is $\text{Cert}_{\text{PV}_i} = b_i \parallel \text{SHA}(b_i)^{d_{\text{PCA}}}$, where $b_i = \text{ID}_{\text{PCA}} \parallel \text{ID}_{\text{PV}_i} \parallel \exp \parallel e_{\text{PV}_i} \parallel N_{\text{PV}_i}$. "TTL" means how long the message is allowed to remain to prevent message flooding. If the key has 1024 bits, the signature has 1024 bits. As exp has 26 bytes [17], the pseudonym certificate length is $L_{\text{Cert}_{\text{PV}}} = 2 + 384 + 26 + 128 + 128 + 128 = 796$ B. Finally, the total message length is $L_{\text{Message}} = 2 + 100 + 4 + 128 + 796 + 1 = 1031$ B, and the pseudonym certificate accounts for 77% of the total message length.

If each message carries a certificate of 796 bytes during the same pseudonym period, the communication overhead is high. If only the first message carries the certificate, the message length is reduced from 1031 bytes to 235 bytes. However, if the first message with the certificate does not arrive at the receiver, other received messages which do not contain the certificate cannot be verified.

Some researchers proposed a mechanism to reduce the communication overhead of secure messages by omitting the inclusion of certificates in messages. Concrete methods such as the periodic omission of certificates, neighbor-based certificate omission, and congestion-based certificate omission were proposed in [9], [18], and [19], respectively. In these schemes, the optimal parameter was obtained by means of simulation. In contrast to the earlier proposals, we found the optimal parameter by means of probability analysis.

Define that α is the number of messages and λ is the number of messages with pseudonym certificate during one pseudonym period. If at least one message with certificate is accepted by the receiver, other arrived messages with the same pseudonym can be authenticated. Define the message authentication probability P_{auth} as $P_{\text{auth}} = 1 - (1 - P_{\text{accept}})^{\lambda}$, where P_{accept} is the packet reception rate. We used a packet reception rate model of broadcast channel in a good channel condition $P(d) = -0.004d + 1/7 \sin(\pi/125d) + 1$ [9], where d is the distance between the sender and the receiver.

Figure 2 shows the relationship among the message authentication probability P_{auth}, the intervehicle distance d, and the number of messages with certificate λ. Assuming that λ is a constant, P_{auth} decreases with the increase of d; assuming that d is a constant, P_{auth} increases with the increase of λ. From the figure, it can be further observed that if $\lambda \geq 5$ and $d \leq 130$, then $P_{\text{auth}} > 95\%$. That means we take $\lambda = 5$, which can meet the need of most vehicles for broadcast message authentication. Thus, $\lambda = 5$ is the optimal number of messages with the pseudonym certificate.

We define *len* as the average message length, and Figure 3 shows the relationship between *len* and α under the condition of $\lambda = 5$. The parameter *len* decreases with the increase of α. It means that during a pseudonym period the more the messages are, the less communication overhead is.

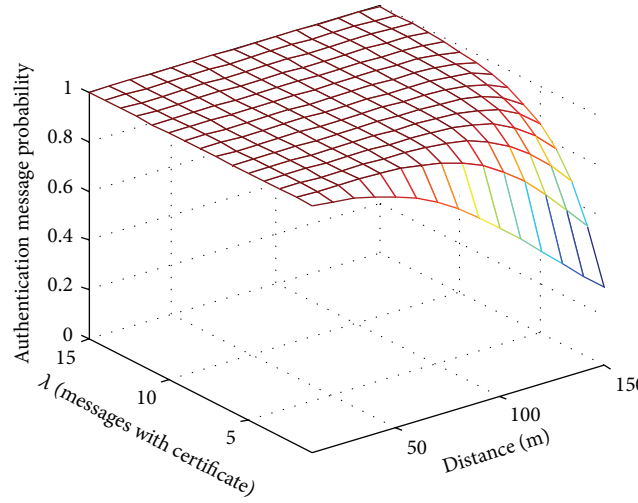


Figure 2: Impact of different d and λ on pseudonymous authentication.

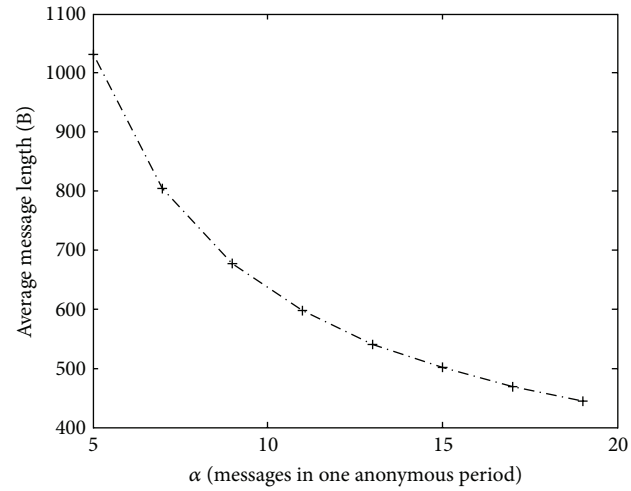


Figure 3: Relationship between average message length and the number of messages during a pseudonym period.

5. Distributed Tracking Protocol

Based on the secret sharing scheme [20], TAs' private key d is assigned to the k authorities, and any one subset of at least m of them can disclose the secret. Generally, there is a trusted center during the initialization stage of the secret sharing scheme. Once the center compromises, the privacy will be leaked. Boneh and Franklin [21] discussed the generation of RSA keys without a dealer, but the protocol required the help of a third party. Cocks [22] presented another protocol to generate shared RSA keys without the help of a third party. There exists a large number of modular exponentiation operations to generate the modulus, and thus the efficiency of the protocol is poor. Malkin et al. [23] extended two parties [21] to k parties.

Instead of time-consuming modular exponentiation, we use modular multiplication to generate N, where N is

the modulus of RSA. And furthermore, we extend the (k, k) threshold schemeto a more general form (m, k), where m is the number of members and k is the threshold value. Based on the above ideas, the distributed tracking protocol is developed. It consists of distributed generation of modulus N, distributed generation of private key and secret share, and collaborative tracking. The first two parts are completed collaboratively by all members during initialization, and the third part is implemented collaboratively by at least m members during the tracking stage.

5.1. Distributed Generation of Modulus N. Each member TA_i $(1 \le i \le k)$ picks random primes p_i, q_i and computes modulus $N = (\sum_{i=1}^{k} p_i)(\sum_{i=1}^{k} q_i)$ without revealing p_i, q_i. Assuming that k is an odd, the steps are as follows:

(1) TA_i picks two random primes p_i, q_i and a random $r_{i,j} \in Z_P$ (prime $P > N$). TA_i computes and sends $r_{i,j}p_i$ and $r_{i,j}q_i$ to TA_j $(1 \le j \le k, j \ne i)$.

(2) TA_j computes $r_{i,j}p_iq_j$ and $r_{i,j}q_ip_j$ and sends $r_{i,j}p_iq_j + r_{i,j}q_ip_j$ to TA_i.

(3) TA_i computes $s_{i,j} = (1/r_{i,j})(r_{i,j}p_iq_j + r_{i,j}q_ip_j) = p_iq_j + p_jq_i$ $(j \ne i)$,

$$\begin{aligned} N_i &= \sum_{j=1, j\ne i}^{k} s_{i,j} + 2p_iq_i \\ &= p_i \sum_{j=1, j\ne i}^{k} q_j + q_i \sum_{j=1, j\ne i}^{k} p_j + 2p_iq_i \end{aligned} \tag{4}$$

and broadcasts N_i in TAs.

(4) All members compute $\sum_{i=1}^{k} N_i$ and get $N = \sum_{i=1}^{k} N_i/2$.

(5) Any member in TAs picks an integer $g \in Z_N^*$ randomly and broadcasts g in TAs.

(6) TA_1 computes and broadcasts $V_1 = g^{N-p_1-q_1+1} \bmod N$; all other members TA_i $(2 \le i \le k)$ compute and broadcast $V_i = g^{p_i+q_i} \bmod N$; all members verify $V_1 = \prod_{i=2}^{k} V_i \bmod N$.

(7) If the verification fails, all members execute steps (1)–(6) again. Otherwise, success is returned.

5.2. Distributed Generation of Private Key and Secret Share. $\phi(N)$ is Euler function of N, denoted as ϕ. If N is the product of two primes,

$$\phi = N - \sum_{i=1}^{k} p_i - \sum_{i=1}^{k} q_i + 1. \tag{5}$$

(1) TA_1 computes $\phi_1 = N - p_1 - q_1 + 1$, and another member TA_i $(2 \le i \le k)$ computes $\phi_i = -p_i - q_i$. Then, each member TA_i $(1 \le i \le k)$ broadcasts $\phi_i \bmod e$ in TAs.

(2) TA_i collects $\phi_j \bmod e$, and computes $\psi = \sum_{j=1}^{k} \phi_j \bmod e = \phi \bmod e$ and $\zeta = -\psi^{-1} \bmod e$.

TA_i obtains its own private key

$$d_i = \left\lfloor \frac{\zeta\phi_i}{e} \right\rfloor. \tag{6}$$

(3) TA_i picks a random degree $m-1$ polynomial $c_i(x) \in Z_P[x]$, satisfying $c_i(0) = d_i$. It computes $s_{i,j} = c_i(\mathrm{ID}_j)$ and sends $s_{i,j}$ to TA_j $(j \ne i)$ secretly.

(4) TA_i collects $s_{j,i}$, then computes the secret share

$$s_i = \sum_{j=1}^{k} s_{j,i} \bmod P. \tag{7}$$

After implementing the protocol, each member obtains the private key d_i and secret share s_i. The operation of computing $\phi \bmod e$ reveals $\log_2 e$ low bits of ϕ. In order to protect ϕ from revealing more bits, we take a small e.

5.3. Collaborative Tracking

(1) The vehicle V^* reports the signed message with a certificate to a tracking authority such as TA_1. TA_1 checks whether the signed message and the certificate are valid.

(2) If passed, TA_1 extracts $\mathrm{ID}_{\mathrm{PV}}$ from $\mathrm{Cert}_{\mathrm{PV}}$ and sends $\mathrm{ID}_{\mathrm{PV}}$ to other members. If $m-1$ members accept a tracking request, they constitute a tracking group. Assume that their identities are $\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_m$.

(3) Participants TA_i $(2 \le i \le m)$ send ${\mathrm{ID}_{\mathrm{PV}}}^{\alpha_i} \bmod N$ to TA_1, where $\alpha_i = s_i l_i(0) \bmod P$.

(4) TA_1 tries all possible u and x $(0 \leqslant u \leqslant k-1, -m < x \leqslant 0)$ and computes

$$\begin{aligned} &{\mathrm{ID}_{\mathrm{PV}}}^{xP+u} \cdot \prod_{i=1}^{m} {\mathrm{ID}_{\mathrm{PV}}}^{\alpha_i} \bmod N \\ &= r \parallel r^{d_v} \bmod N_V \parallel (\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_{\mathrm{V}} \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}}. \end{aligned} \tag{8}$$

It extracts $(\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}}$ and further gets $\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp$ with e_{PCA}. TA_1 gets ID_V according to the actual meaning of the strings.

Then TA_1 submits ID_V to PCA, and PCA puts it into the blacklist. PCA will reject the request of the vehicle in the blacklist for pseudonym certificates. So, the pseudonym tracking protocol combined with the pseudonym issuance protocol can realize the revocation of malicious vehicles.

In Lin et al.'s protocol [7], a centralized method for tracking is adopted. A trace manager (TM) computes a vehicle's private key from a signed message in order to disclose the vehicle's identity. Once TM is compromised, a large amount of privacy information is leaked.

In Schaub et al.'s protocol [14], the secret sharing method for tracking is adopted to prevent misuse and abuse of a system. The method is distributed, but it generally requires a trusted center to distribute secret share in the initialization

phase. Therefore, the trusted center will be a secure bottleneck.

In our tracking protocol, all the processes in the secret sharing method adopt a fully distributed structure, such as generation of private key and secret share, generation of modulus N and collaborative tracking. As a result, our tracking protocol avoids a single point of failure.

6. Analysis

6.1. Security Analysis

Proposition 1. *Neither issuing authorities nor other vehicles can determine the relationship between a pseudonym and an identity in the protocol family, so the scheme achieves anonymity.*

Proof. (1) During the issuance stage, PCA gets $c_i = (x_i^{e_{\mathrm{PCA}}}) \cdot \mathrm{SHA}(b_i) \bmod N_{\mathrm{PCA}}$. V picks x_i randomly, so PCA cannot get b_i and $\mathrm{ID}_{\mathrm{PV}_i}$. Though PCA knows ID_V, it cannot establish the relationship between ID_V and $\mathrm{ID}_{\mathrm{PV}_i}$.

(2) Other vehicles get the signed messages and extract $\mathrm{ID}_{\mathrm{PV}_i} = (r_i \parallel r_i^{d_v} \bmod N_V \parallel (\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}})^e \bmod N$. They do not know TAs' private key d and thus cannot obtain V's identity ID_V. □

Proposition 2. *The m authorities in TAs can disclose the identity ID_V from a pseudonym ID_{PV}, so the scheme achieves traceability.*

Proof. Let $\psi = \phi \bmod e = \sum_{i=1}^{k} \phi_i \bmod e$, $\zeta = -\psi^{-1} \bmod e$, and then $\zeta\phi + 1 = -\phi^{-1}\phi + 1 = 0 \bmod e$, indicating $e \mid \zeta\phi + 1$.

Since $de = 1 \bmod \phi$, this gives $d = (\zeta\phi+1)/e = (\sum_{i=1}^{k} \zeta\phi_i + 1)/e$. Let $d_i = \lfloor \zeta\phi_i/e \rfloor$; then

$$d = \sum_{i=1}^{k} d_i + u \quad (0 \le u \le k-1). \tag{9}$$

Construct a degree $m-1$ polynomial $c(x) = \sum_{j=1}^{k} c_j(x) \bmod P$, obviously satisfying $c(0) = \sum_{j=1}^{k} c_j(0) = \sum_{j=1}^{k} d_j$ and $c(\mathrm{ID}_i) = \sum_{j=1}^{k} c_j(\mathrm{ID}_i) = \sum_{j=1}^{k} s_{j,i} = s_i$. According to the Lagrange interpolation formula $\sum_{j=1}^{k} d_j = \sum_{i=1}^{m} s_i l_i(0) \bmod P$, let $\alpha_i = s_i l_i(0) \bmod P$, so

$$\sum_{j=1}^{k} d_j = \sum_{i=1}^{m} \alpha_i + xP. \tag{10}$$

Since $0 \le \sum_{j=1}^{k} d_j < \phi$, $0 \le \alpha_i < P$, $0 \le \sum_{i=1}^{m} \alpha_i < mP$, therefore $-m < x \le 0$.

Taking the formula (10) into the formula (9), we can obtain the following formula:

$$d = \sum_{i=1}^{m} \alpha_i + xP + u \quad (0 \le u \le k-1,\ -m < x \le 0). \tag{11}$$

Consider the issuance protocol and obtain

$$\begin{aligned} & r \parallel r^{d_v} \bmod N_V \parallel (\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}} \\ &= \mathrm{ID}_{\mathrm{PV}}{}^{d} = \mathrm{ID}_{\mathrm{PV}}{}^{\sum_{i=1}^{m} \alpha_i + xP + u} \\ &= \mathrm{ID}_{\mathrm{PV}}{}^{xP+u} \cdot \prod_{i=1}^{m} \mathrm{ID}_{\mathrm{PV}}{}^{\alpha_i} \bmod N. \end{aligned} \tag{12}$$

Compute $((\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}})^{e_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}}$ with e_{PCA} and obtain V's identity ID_V. □

Proposition 3. *Regardless of the PCA, the vehicles and the outside attacker, forging the pseudonym certificate is as difficult as solving a large integer factorization problem.*

Proof. A pseudonym $\mathrm{ID}_{\mathrm{PV}_i}$ is coproduced by V and PCA, because $r_i \parallel r_i^{d_v} \bmod N_V$ is provided by V and $(\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_V \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}}$ is provided by PCA.

(1) PCA cannot provide $r_i \parallel r_i^{d_v} \bmod N_V$ to forge a pseudonym.

(2) V cannot provide $(\mathrm{ID}_{\mathrm{PCA}} \parallel \mathrm{ID}_{V^*} \parallel \exp)^{d_{\mathrm{PCA}}} \bmod N_{\mathrm{PCA}}$ to forge the pseudonym of V^*.

(3) An external attacker without V's and PCA's private keys cannot pass the authentication, because the issuance protocol is with two-way authentication. Therefore, the attacker neither obtains the blind signature nor personates PCA to sign the pseudonym certificate. □

In short, forging an RSA signature or cracking an RSA cipher is as difficult as factorizing a large integer, so the scheme achieves unforgeability.

Proposition 4. *The scheme is robust.*

Proof. (1) The separation of pseudonym issuance authorities and tracking authorities, to some extent, reduces the risk.

(2) Some PCAs are deployed in the model. Once a PCA fails, other PCAs can still provide pseudonym issuance service.

(3) In the pseudonymous authentication protocol, the optimal number of messages with the pseudonym certificates is suggested. It ensures the message authentication probability $P_{\mathrm{auth}} > 95\%$. That means when a message arrives, a vehicle can verify the signature with high probability. Therefore, the protocol not only reduces the communication overhead but also ensures robustness.

(4) During the TAs initialization stage, the generation of modulus N and the generation of the private key and the secret share are distributed fully. So, the scheme does not require a trusted center, and it avoids any single point of failure.

(5) During the TAs operation stage, as long as the number of compromised members is not more than $m-1$, privacy cannot be leaked. □

We further compared our scheme with similar works that are intended to ensure conditional privacy preserving communication [7, 14]. The results of comparisons of security features among our scheme, Lin et al.'s scheme [7], and

Table 1: Security features comparisons.

	Anonymity	Traceability	Authentication	Unforgeability (especially against authority attacks)	Robustness (especially for tracking)
Lin et al.'s scheme [7]	Yes	Yes	Yes	No	Centralized
Schaub et al.'s scheme [14]	Yes	Yes	Yes	No	Distributed
Our scheme	Yes	Yes	Yes	Yes	Fully distributed

Table 2: Computation cost comparisons.

	Issuance	Signature and verification	Tracking
Lin et al.'s scheme [7]	T_{exp}	$12T_{\text{exp}} + T_P + 12T_{\text{exp}} + 2T_P$	$2T_{\text{exp}}$
Schaub et al.'s scheme [14]	$7T_e + 4T_d + (T_e + T_d)/n$	$T_e + T_d$	T_d
Our scheme	$6T_e + 3T_d + 3(T_e + T_d)/n$	$T_e + T_d$	$(k + 1)\,T_d$
Computation time (ms)			
Lin et al.'s [7]	0.6	27.9	1.2
Schaub et al.'s [14]	$17.4 + 4.2/n$	4.2	4
Ours	$13.2 + 12.6/n$	4.2	$4(k + 1)$

Schaub et al.'s scheme[14] are shown in Table 1. All the three schemes provide anonymity, traceability, and authentication.

Lin et al.'s scheme is based on a group signature method, in which a member manager (MM) generates member private keys and sends them privately; MM knows all private keys, and it can forge a valid group signature on an arbitrary message. Schaub et al.'s scheme adopts a blind signature method to issue certificates, and thus CA or PP does not know the relationship of an identity and a pseudonym; unfortunately, CA or PP can forge a pseudonym certificate for itself. Therefore, Lin et al. and Schaub et al.'s schemes are not secure against authority forge attacks. In our scheme, a pseudonym is coproduced by a manager and a vehicle; neither PCA nor V is capable of providing complete data to forge a pseudonym; our scheme is unforgeability, especially against authority forge attacks.

As mentioned in Section 5, the tracking methods for Lin et al., Schaub et al. and ours are centralized, distributed, and fully distributed, respectively. Our scheme has better robustness.

6.2. Performance Analysis

6.2.1. Computation Overhead. For convenience to evaluate the computation cost of the protocol, we ignored the computation cost of some operations such as a hash function and a multiplication operation, since they are quite light in terms of load. We focused on some time-consuming operations defined in the following notations.

T_P: The time of executing a bilinear map operation.

T_{exp}: The time of executing a modular exponentiation operation in the cyclic group.

T_e: The time of executing RSA encryption or RSA verification.

T_d: The time of executing RSA decryption or RSA signature.

In order to provide the precise comparisons of computation cost, we use the experiment data in [24–26] to evaluate them. The experiment environment is operated on a standard PC, whose processor is Pentium IV with the maximum clock speed of 3 GHz. The pairing system is considered the Tate pairing system. The order of a nonsupersingular curve over a finite field $E(F_p)$ is 160 bits, which is as difficult to break as 1024-bit RSA. In this experiment environment, it requires 4.5 ms to perform a bilinear map operation and 0.6 ms to perform a modular exponentiation operation [25, 26]. It requires 0.2 ms to perform RSA encryption/verification and 4 ms to perform RSA decryption/signature [24].

Table 3: Communication cost comparisons.

	Issuance (rounds)	Signed messages (B)	Tracking (B)
Lin et al.'s scheme [7]	1	289	309
Schaub et al.'s scheme [14]	4	1031	$512 + 519/m$
Our scheme	3	$235 + 796\,\lambda/\alpha$	$512 + 519/m$

The results of comparisons of computation cost are shown in Table 2, where n is the number of pseudonym certificates obtained at one time and k is the number of tracking authorities. Some common parameters and secret keys are generated in the system initialization, and for convenience we did not evaluate the computation cost of the initialization in all the three schemes. In Schaub et al.'s scheme and ours, the average computation costs for issuance and tracking are considered.

Compared with Lin et al.'s scheme, ours and Schaub et al.'s scheme require less computation for signature and verification and more computation for issuance and track. Compared with Schaub et al.'s, our scheme requires less computation for issuance if $n > 2$ and more computation for tracking.

6.2.2. Communication Overhead. In Table 3, α is the number of signed messages and λ is the number of signed messages with pseudonym certificate during one pseudonym period in

the authentication protocol; m is the threshold value in the distributed tracking protocol. The communication cost of Lin et al.'s scheme is the lowest, but their scheme suffered from the private key revealing attacks, in which MM knows the private key of each member. Schaub et al.'s scheme and ours rely on the blind signature method and achieve vehicular privacy protection in the presence of the authority. Compared with Schaub et al.'s scheme, our scheme is efficient in terms of the communication overhead.

6.2.3. Storage Overhead. In Schaub et al.'s scheme and ours, the storage cost of the vehicle is high because some pseudonym certificates need to be stored; the storage cost of the manager for tracking is very little because an identity can be obtained directly from the pseudonym certificate. On the contrary, the storage cost of the manager in Lin et al.'s scheme is high because the record set (A_i, ID_i) needs to be stored.

7. Conclusions

In this paper, a secure and efficient pseudonym management scheme for vehicular ad hoc networks is proposed. The scheme not only maintains the property of conditional privacy preservation but also provides the advantages in security against authority forge attacks and better robustness. In the scheme, a pseudonym is coproduced by V and PCA to avoid the deception of either party. A blind signature method is used to achieve the separation of issuance and tracking. Based on the improved share generation scheme of the RSA keys, the distributed tracking protocol is proposed to avoid a single point of failure. By searching for the optimal number of messages with a pseudonym certificate, the efficient pseudonym authentication mechanism is given to reduce communication overhead. By uniting the pseudonym issuance protocol and the tracking protocol, malicious vehicles are revoked easily. Moreover, compared with Schaub et al.'s scheme, the communication cost and computation cost in our scheme are lower. As a result, our proposed scheme is suitable for anonymous communication with tracking requirements in VANETs, since it provides security, robustness, and efficiency.

For future research, we will discuss interdependencies of various factors, establish systematic evaluation mechanism of the overall performance, and further enhance the performance.

Acknowledgments

The authors acknowledge the financial support of the National Natural Science Foundation of China (no. 60873195), the National High Technology Research and Development Program ("863" Program) of China (no. 2011AA060406), and the Natural Science Foundation of Anhui Province (no. 090412051). The authors are grateful for the anonymous referee for the careful checking and helpful comments that improved this paper.

References

[1] J. Blum, A. Eskandarian, and L. Hoffman, "Challenges of intervehicle adhoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 347–351, 2004.

[2] J. P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security and Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[3] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proceedings of the 4th Workshop on Hot Topics in Networks (HotNets-IV)*, pp. 1–6, College Park, Md, USA, November 2005.

[4] S. Zeadally, R. Hunt, Y. S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[5] M. Raya and J. P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks (SASN '05)*, pp. 11–21, Alexandria, Va, USA, November 2005.

[6] M. Raya and J. P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[7] X. Lin, X. Sun, P. H. Ho, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[8] G. Calandriello, P. Papadimitratos, J. P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET '07)*, pp. 19–28, Montreal, Canada, September 2007.

[9] G. Calandriello, P. Papadimitratos, J. P. Hubaux, and A. Lioy, "On the performance of secure vehicular communication systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 898–912, 2011.

[10] R. Lu, X. Lin, H. Zhu et al., "ECPP: eficient conditional privacy preservation protocol for secure vehicular communications," in *Proceedings of the IEEE 27th Conference on Computer Communications (INFOCOM '08)*, pp. 1229–1237, Phoenix, Ariz, USA, 2008.

[11] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, "A scalable robust authentication protocol for secure vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1606–1617, 2010.

[12] Y. Hao, Y. Cheng, C. Zhou, and W. Song, "A distributed key management framework with cooperative message authentication in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 616–629, 2011.

[13] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Crypto '04*, vol. 3152 of *Lecture Notes in Computer Science (LNCS)*, pp. 41–55, 2004.

[14] F. Schaub, F. Kargl, Z. Ma, and M. Weber, "V-tokens for conditional pseudonymity in VANETs," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, Sydney, Australia, April 2010.

[15] P. Papadimitratos, L. Buttyan, T. Holczer et al., "Secure vehicular communication systems: design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, 2008.

[16] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Crypto '82*, pp. 199–203, Plenum, 1982.

[17] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X. 509 Public Key Infrastructure Certificate and CRL Profile," 2011, http://www.ietf.org/rfc/rfc2459.txt .

[18] E. Schoch and F. Kargl, "On the efficiency of secure beaconing in VANETs," in *Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec '10)*, pp. 111–116, New York, NY, USA, March 2010.

[19] M. Feiri, J. Petit, and F. Kargl, "Congestion-based certificate omission in VANETs," in *Proceedings of the 9th ACM international workshop on Vehicular Inter-Networking, Systems, and Applications*, pp. 135–138, 2012.

[20] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[21] D. Boneh and M. Franklin, "Efficient generation of shared rsa keys," in *Proceedings of Crypto '97*, pp. 425–439, 1997.

[22] C. Cocks, "Split knowledge generation of rsa parameters," in *Proceedings of the 6th IMA International Conference on Cryptography and Coding*, pp. 89–95, 1997.

[23] M. Malkin, T. Wu, and D. Boneh, "Experimenting with shared generation of RSA keys," in *Proceedings of the Internet Society's 1999 Symposium on Network and Distributed System Security*, pp. 43–56, San Diego, Calif, USA, 1999.

[24] OpenSSL, "The Open Source Toolkit for SSL/TLS," 2012, http://openssl.org/.

[25] M. Scott, "Efficient implementation of cryptographic pairings," 2007, ftp://ftp.disi.unige.it/pub/.person/MoraF/CRYPTO/PARING/mscott-samos07.pdf.

[26] L. Chen, S. L. Ng, and G. Wang, "Threshold anonymous announcement in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 605–615, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 469076, 10 pages
http://dx.doi.org/10.1155/2013/469076

Research Article

A PEFKS- and CP-ABE-Based Distributed Security Scheme in Interest-Centric Opportunistic Networks

Fei Wang,[1,2] YongJun Xu,[1] Lin Wu,[1,2] Longyijia Li,[3] Dan Liu,[3] and Liehuang Zhu[3]

[1] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[2] *University of Chinese Academy of Sciences, Beijing 100049, China*
[3] *School of Computer Science & Technology, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Fei Wang; wangfei@ict.ac.cn

Received 26 December 2012; Accepted 16 March 2013

Academic Editor: Hongsong Zhu

Security is a crucial issue in distributed applications of multihop wireless opportunistic network due to the features of exposed on the fly communication, relaxed end-to-end connectivity, and vague destinations literately. In this paper, we focus on problems of user privacy leakage and end-to-end confidentiality invasion in content-based or interest-centric wireless opportunistic network. And we propose a public-encryption-with-fuzzy-keyword-search- (PEFKS-) and ciphertext-policy-attribute-based-encryption- (CP-ABE-) based distributed security scheme by refining and compromising two-pairing-based encryption, searchable encryption, and attribute-based encryption. Our scheme enables opportunistic forwarding according to fuzzy interests preserving full privacy of users and ensures end-to-end confidentiality with a fine-grained access control strategy in an interest-centric scenario of large-scale wireless opportunistic networks. Finally, we analyze and evaluate the scheme in terms of security and performance.

1. Introduction

Opportunistic network is a type of ad hoc wireless network which has common features of delay tolerant network (DTN), which achieves routing through opportunities of meeting between mobile nodes. It aims to solve the problem of communication in the presence of intermittent network connectivity. And to this end, opportunistic network (OPPNET) has the following features.

(i) Communication is on the fly, thus exposed to all powerful adversary parties trying to spy the privacy of users and steal the information.

(ii) Forwarding decisions are made on the fly based on any possible information from a collapsed network architecture [1], owing to the aim of transmitting a message over any communication gaps.

(iii) Only vague destinations literately exist because a message ought to be sent to a group of nodes according to some principles: deployed near a specific location, equipped with same sensors or actuators, or interested in specific information [2].

Content-based opportunistic network uses content of message which concerns users' interests to make forwarding decisions, and a most popular application scenario is that network nodes which are actually pedestrians look forward to acquiring and sharing information through opportunistic communication with others. Every pedestrian carries one or more portable wireless terminals such as smart phones, tablets, and laptops. More precisely, it is people-centric or interest-centric opportunistic network. In such a scenario, a user can be a subscriber with interests in some topics; he could also be a publisher intending to publish contents about some topics. Security in such type of opportunistic network is a crucial issue. We consider two components of security in opportunistic network: privacy and confidentiality.

First is privacy. In a content-based opportunistic network, interest advertisements from subscribers and published contents from publishers both need to be forwarded based on the interest-oriented information (maybe some indexes, abstracts, keywords, etc.) which are contained in the messages. The routing could be multihop through several intermediate nodes which might not be trusted by subscribers or publishers. The subscribers do not want other

subscribers and intermediate nodes to know their interests. The situation is almost the same as publishers: they do not want other publishers, intermediate nodes, or subscribers that are not designated to obtain the interest orientations of their published contents. However, in precedent research about interest-centric network, forwarding relies on explicit queries or indexes, which leads to possible threat of privacy leakage due to matching between message indexes and users' identities.

Second is end-to-end confidentiality which is considered as a fundamental security requirement. Publishers not only want to impose a ban on the access to the plaintext payload of their published contents, but also want to attach a precise access control strategy to their every message so that only the designated subscribers who have certain credentials or attributes can access the payload. For example, when top layer of some IT company wants to publish a product fault survey, they may make such an access control structure as shown in Figure 1 which means only the department manager of Technology Department or Quality Control Department, staff with management level over 5 or a consultant called Charlie Eppes, could access the survey. However, the reasons of no stable end-to-end connection and group target of a message not only make traditional end-to-end encryption unsuitable, but also increase the difficulty of fine-grained access control on shared data.

In this paper, we propose a *PEFKS*- and *CP-ABE*-based distributed security scheme in interest-centric opportunistic network.

The main contributions of this paper are as follows.

(i) We design a *PEFKS*-based privacy protecting forwarding decision scheme, which on one hand enables subscribers to publish encrypted fuzzy interests and on the other hand enables intermediate nodes to forward messages according to encrypted fuzzy interests. The scheme ensures users' full privacy in the circumstance of opportunistic network and enhances anonymity through fuzzy interests. To the best of our knowledge, we are the first to enable partial match on fuzzy interests in opportunistic network.

(ii) We embed the concept of attribute-based identity into opportunistic network to adapt to the feature of no explicit destinations. Then we design a *CP-ABE*-based confidentiality protecting scheme, in which publishers make and attach an expressive access control strategy to the messages they are about to send. Subscribers whose attribute-based identities satisfy the access control strategy are legalized to decrypt the ciphertext, finally achieving confidentiality with a fine-grained access control strategy on shared data.

(iii) We implement and analyze the security and performance of the schemes and verify the feasibility of our security schemes.

2. Related Work

Related research work is still scarce because security in content-based opportunistic network is a quickly emerging problem and most of the security schemes existing in Internet, wireless sensor networks (WSNs), mobile ad hoc networks (DTN, or MANET) are not suitable.

As far as we are concerned, Lilien et al. [3] were the first to consider security in opportunistic network. They proposed several challenges in privacy and confidentiality of opportunistic network in particular the requirement for end-to-end confidentiality, but they did not propose any possible security solution. What is more, they did not analyze the issue of context privacy or content privacy.

Nguyen et al. proposed a probabilistic routing protocol for ICMAN (intermittently connected mobile ad hoc network) in [4], which indicates the very first idea to protect privacy and confidentiality. In their protocol, if senders want to send messages to receivers, it hashes all the values of message head. Before intermediate node does the partial match, it first calculates its attributes using the same hash functions. Only hash function is used hereby to achieve a relatively computational efficiency, but it is obviously prone to dictionary attack. As for confidentiality, they used the information (evidence/values) that the sender knows about the destination node as keys, so only the destination node can decrypt the cipher messages to get the plain messages in a community scenario. In this scheme the confidentiality is based on assumption that all in-community members can be trusted which obviously cannot be guaranteed in opportunistic network.

In the neighboring area of DTN, a bundle security protocol (BSP) [5] was defined to enhance the security of communications in DTN. In BSP, a confidentiality block is included to enable the encryption of the entire payload at the source and the decryption at the final destination based on the identifier of the destination. It does not enable encryption based on the interests of destinations or partial matches that can be used to make interest-centric forwarding decisions. Those features are thus not enough to satisfy the interest-centric scenario.

Shikfa et al. proposed a scheme for content-based and context-based opportunistic network in [1, 6]. In content-based scenario, they defined a three-level privacy model and two security primitives "secure look-up" and "setup of forwarding tables" and proposed a distributed security scheme based on multiple layer commutative encryption (MLCE) in which r-hop neighbor nodes share keys to encrypt and decrypt messages. There are two disadvantages: one is that only single-word keyword is supported, it is not flexible, and the other is that the number of shared keys will explode when topology changes frequently. In context-based scenario, they replaced identities of classic identity-based encryption (IBE) with attributes contained in context to assure end-to-end confidentiality. Also, they used improved public encryption with keyword search (PEKS) to make forwarding decisions to protect users' privacy. Their scheme is flexible enough to meet the privacy requirements of context-based forwarding. But it only supports strictly limited attributes arrange rules such as ⟨[Mail], [Workplace], [Status]⟩ which is not suitable for content-based forwarding. In addition, the sender cannot apply a fine-grained access control strategy to shared data.

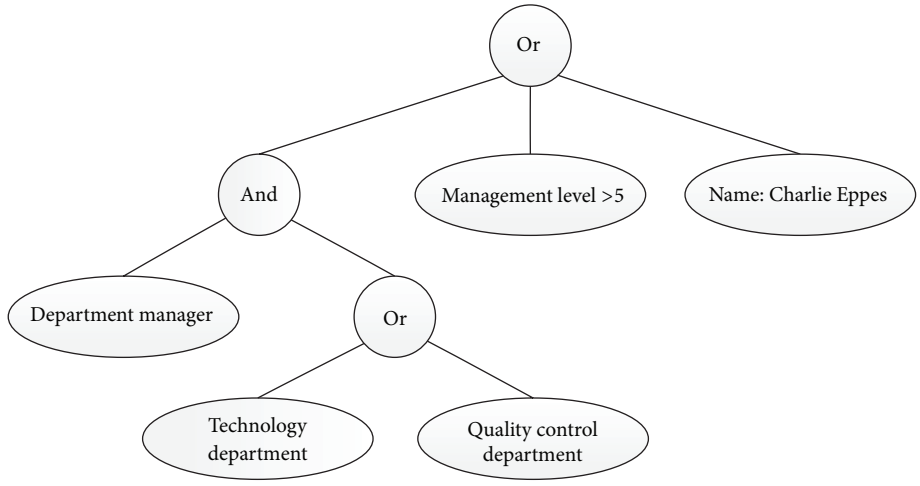


Figure 1: Access policy, for example, in introduction.

There are also other researches concerning security in opportunistic network. As for trust relationship, Li and Das designed a trust-based framework in [7] to more accurately evaluate an encounter's competency met, which can be flexibly integrated with a large family of existing data forwarding protocols. The proposed framework was implemented with PROPHET, demonstrating great effectiveness against "black hole" attacks. But in [7], neither privacy nor confidentiality is considered. Shin et al. presented and implemented AnonySense, a privacy-aware system for realizing pervasive applications based on collaborative, opportunistic sensing by personal devices in [8]. In their system, the sensing tasks and sensor data can be anonymized through the collaboration of Tor network, MIX network and anonymization service (AS) before being reported back. Such a framework achieves great flexibility with users' privacy respected, yet the privacy in opportunistic forwarding is not considered. To improve privacy and confidentiality in opportunistic network, we should mine deeper in modern cryptography.

PEKS was presented by Boneh et al. [9] based on bilinear pairings which makes searching on encrypted data possible. PEKS was improved and used in [1]. But only explicit keywords are not enough to fit in complex application requirements, especially when users know little about the network. When users submit complete queries, they will feel "left in the dark" [10] and have to use a try-and-see approach for finding information. In this respect, fuzzy interests and match of them which can enhance the user interactivity are needed, which current PEKS implemented in opportunistic network cannot achieve. In [11], wildcard-based fuzzy set construction was introduced, but for a word of length l and similarity d quantized by edit distance [12], the size of fuzzy set is $O(l^d)$. In this fuzzy set, many meaningless words that existed bring spatial redundancy. Later, in [13], dictionary-based fuzzy set construction was proposed to improve the efficiency dramatically.

Shikfa introduced IBE and their scheme fits well in context-based opportunistic network. IBE is proposed by Shamir [14] and Boneh and Franklin [15] based on bilinear pairings. It enables resource providers (publishers in opportunistic network) to utilize user's identity as public key without querying for public key certificate online. This feature fits the relaxed-connection environment of opportunistic network very well and was demonstrated in Shikfa's work.

Based on IBE, Sahai and Waters proposed attribute-based encryption or fuzzy identity-based encryption (*ABE*) in 2005 [16]. Research in *ABE* has been hot since then. Su et al. compared *ABE* with IBE and summed up for advanced features in [17]. All these features are very suitable for opportunistic networks. The first is that *ABE* enables resource providers to encrypt messages with only attributes without considering the number or identities in the group, which can reduce the encryption cost on opportunistic network nodes and protect users' privacy in a tailor-made no-explicit-destination environment of opportunistic network. Secondly, only the one whose attribute-based identity satisfying the requirement of cipher text can decrypt the cipher text, which assures end-to-end confidentiality in a relaxed connected environment of opportunistic network. Third is that in *ABE*, users' key was related to random polynomial or random number; thus no collusion is possible among opportunistic network users. The last is that flexible access control strategies such as the AND, OR, NOT, and THRESHOLD of attributes are supported which will dramatically increase the flexibility. The first basic *ABE* [16] scheme only supports threshold access strategy; then researchers developed it and proposed key-policy attribute-based encryption (*KP-ABE*) [18] and ciphertext-policy attribute-based encryption (*CP-ABE*) [19] to achieve more flexible access control strategies. The former enables users to make rules about the messages they are going to receive, and the latter allows senders to make access strategies for ciphertexts.

From the aforementioned work, it could be concluded that none of related research work has achieved privacy or confidentiality with good flexibility in content-based opportunistic network. Some kind of improved PEKS and *ABE* can be our first choice. Therefore we propose a privacy protecting forwarding decision scheme based on *PEFKS* and a confidentiality protecting scheme based on *CP-ABE*.

3. Problem Statement

3.1. Reference Model. The involved nodes of interest-centric opportunistic network are some people working in the same or near places, carrying portable devices with one or more wireless communication interfaces such as Wi-Fi, near field communication (NFC), and Bluetooth. They devote themselves to the opportunistic network system to get information (news, e-mails, gossip, etc.) through multi hop forwarding based on their opportunistic mobility. Their roles in such a system are equal and could work as a subscriber, an intermediate node, and a publisher [20] at the same time. In fact, all users are supposed to be greedy and intend to get without sharing. Therefore, some incentive system exists aiming at leading users to offer their devices' capacity such as storage space and computing power.

Figure 2 depicts the reference model of opportunistic communication in previously mentioned application scenario, in which a subscriber A broadcasts its interest advisements RA, while a publisher C publishes contents PC. Intermediate nodes are responsible for two things: setup of forwarding tables FT according to interest advisements and decision making based on forwarding tables.

To specify the limitation of interest keywords, D_0 is defined as a dictionary containing all valid English words which might be chosen as candidate interests. Every valid user has a subset of candidate keywords W_u^* which would be constant in the lifetime of an opportunistic network. Additionally, all people involved have their own attributes set $Y = \{y_{u,1}, y_{u,2}, \ldots, y_{u,n}\}$ to describe their identities. Users' interests vary over time with the change of hot topics. So interest-centric opportunistic network enables users to update their current interests which all come from W_u^*.

Subscriber's interest advertisement message is composed of two parts: control information CI_A of the message and identity ID_A of A, expressed as $RA = [CI_A, ID_A]$. Without considering security of the network, CI_A is a sub set of W_A^*. ID_A is identity of subscriber A.

Publisher's published message is composed of three parts: control information, payload, and access policy, which could be represented as $PC = [CI_C, P_C, A_{P_C}]$. CI_C represents publisher-defined keywords which can be used as an index of the payload P_C. A_{P_C} is access policy of P_C which supports AND, OR, and THRESHOLD of interests. In this paper we refer to the definition of access policy from [19] directly.

An intermediate node's kth record in forwarding table is composed of two parts: routing information and set of identities of subscribers who are interested in the routing information. We express it like $FT[k] = [RI_k, SID_k]$. RI_k is the routing information and SID_k is the set of identities of subscribers who are interested in the routing information. What is more, every intermediate node will maintain a message list $List_B = \{PC_{B,1}, PC_{B,2}, \ldots, PC_{B,L_B}\}$ containing the messages to be forwarded with the size of L_B.

3.2. Threat Model. In this paper, we focus on two kinds of threats: one is user privacy leakage through indexing information embedded in interest advertisements and published contents, and the other is information stealing from published contents. The threat model is illustrated in Figure 3.

The first kind of threats aims at acquiring and recording the trends of users' interests and published contents which are considered as privacy. The executors could be an adversary or a malicious authenticated user. Here we suppose that it is extremely hard to pass the authentication for an adversary; thus neither impersonation nor forgery is possible. As formor type, the main attacking measure is eavesdropping and brute force attack. Adversary, whose computation capability is powerful, would sniff all the packets in its communication range and try to reveal the privacy. As the latter, executors have normal computation power and act just like the normal users by advertising, forwarding, and publishing. What is more, they would advertise or publish some specific topics to confuse and pry into privacy.

The second kind of threats could also be performed by the afoementioned two kinds of executors. An adversary would keep attacking with brute force, and malicious users would advertise interests as many as possible on purpose and try to find some clues of personal information.

3.3. Design Goals. There are two goals we would like to achieve in our research.

(i) We intend to design a privacy preserving forwarding decision scheme, which on one hand enables subscribers to publish encrypted fuzzy interests on the other hand enables intermediate nodes to forward messages according to encrypted fuzzy interests. The scheme can assure users' full privacy in the circumstance of opportunistic network and enhance anonymity through fuzzy interests.

(ii) We want to embed the concept of attribute-based identity into opportunistic network to adapt to the feature of no explicit destinations. And we intend to design a confidentiality preserving scheme with a fine-grained access control strategy on shared data, in which publishers make and attach an expressive access control strategy to the messages they are about to send. Only subscribers whose attribute-based identities satisfy the access control strategy are legalized to decrypt the ciphertext.

4. Proposed Scheme

In this chapter we will firstly introduce the security preliminaries needed in the demonstration of our scheme in Section 4.1. Then our *PEFKS*- and *CP-ABE*-based distributed security scheme will be described in Sections 4.2 and 4.3.

4.1. Preliminaries

(1) PEKS will be used in in forwarding decision scheme to protect privacy. It consists of three preliminaries: PEKS, Trapdoor, and Test.

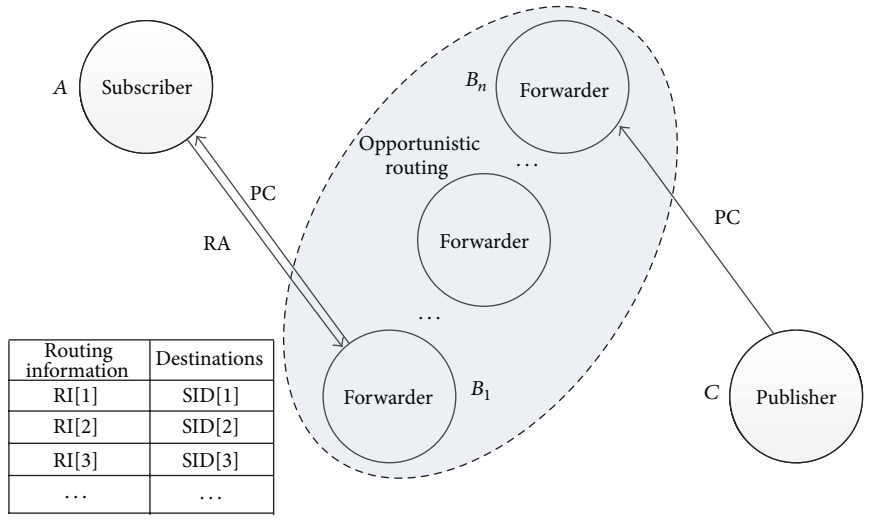


Figure 2: Opportunistic network application scenario model.

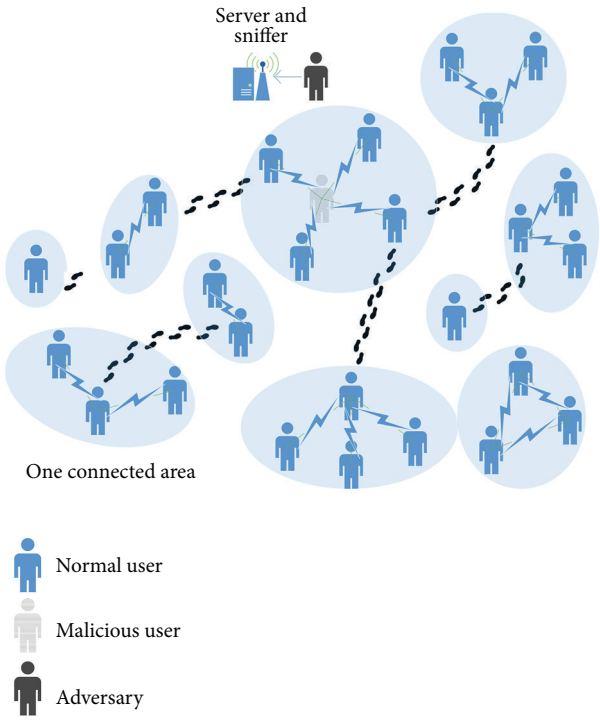


Figure 3: Threat model of the working people application scenario.

(i) *PEKS*: it inputs public key of some node A and a keyword w, outputs a searchable encrypted keyword w', w' cannot be obtained only with w:

$$w' = PEKS\left(A_{\text{pub}}, w\right). \tag{1}$$

(ii) *Trapdoor*: it inputs private key of some node A and a keyword w and outputs the trapdoor TD_w of w. TD_w is private because computation of it needs private key:

$$TD_w = Trapdoor\left(A_{\text{priv}}, w\right). \tag{2}$$

(iii) *Test*: it inputs a searchable encrypted keyword w' and a trapdoor D, outputs *true* if and only if TD is the trapdoor of w, otherwise outputs *false*:

$$Flag = Test\left(w', TD\right). \tag{3}$$

(2) *CP-ABE* will be used in the confidentiality preserving scheme to realize a fine-grained access control strategy. It consists of four preliminaries: *CPA_Setup*, *CPA_Encrypt*, *CPA_KeyGen*, and *CPA_Decrypt*.

(i) *CPA_Setup*: setup of *CP_ABE* needs a third party authority. Firstly let G_1 be a bilinear group of prime order p, and let g be a generator of G_1. Secondly choose two random components $\alpha, \beta \in Z_p$ and then publish the public key and master key:

$$CPA_{\text{pub}} = \left\{G_1, g, h = g^{\beta}, f = g^{1/\beta}, e(g,g)^{\alpha}\right\}, \tag{4}$$
$$CPA_{\text{master}} = \left\{\beta, g^{\alpha}\right\}.$$

(ii) *CPA_Encrypt*: sender makes the access policy T_M for the message about to be sent, encrypts plain text M, and gets cipher text CT:

$$CT = CPA\_Encrypt\left(CPA_{\text{pub}}, M, T_M\right). \tag{5}$$

(iii) *CPA_KeyGen*: the receiver submits its attributes set S to authority, and authority uses CPA_{master} and S to compute the private key $CPA_{\text{priv},S}$:

$$CPA_{\text{priv},S} = CPA\_KeyGen\left(CPA_{\text{master}}, CPA_{\text{pub}}, S\right). \quad (6)$$

(iv) *CPA_Decrypt*: in the end, receiver uses $CPA_{\text{priv},S}$ to decrypt CT if S satisfies T_M, receiver will get M.

$$M = CPA\_Decrypt\left(CT, CPA_{\text{priv},S}, T_M\right). \quad (7)$$

4.2. PEFKS-Based Privacy Preserving Forwarding Scheme

(1) *Initialization*. The initialization is completed by the trusted third party TTP. It possesses a public key TTP_{pub} and a private key TTP_{priv}. It should be noticed that, TTP is not always on-line in lifetime of opportunistic network; in fact, it only works when a new user registers to participate. As for *PEFKS*, TTP will distribute its TTP_{pub} to every user registering. When a subscriber A_i submits its candidate interests set $W^*_{A_i}$ to TTP, TTP constructs a fuzzy set for every word in $W^*_{A_i}$ using dictionary-based fuzzy set construction [13]:

$$F^{D_0}_{w,d} = \left\{w', ed\left(w', w\right) \le d,\ w' \in D_0\right\}, \quad \text{for each } w \text{ in } W^*_{A_i}. \quad (8)$$

Then trusted third party (TTP) constructs trapdoor sets for all fuzzy sets using private key of TTP and returns them to A_i:

$$T_{A_i,w} = \left\{TD = Trapdoor\left(TTP_{\text{priv}}, w'\right), \text{for each } w' \text{ in } F^{D_0}_{w,d}\right\}. \quad (9)$$

After the registration, TTP can be off-line for some time. There may be a doubt why TTP cannot always be on-line to process the subscriber's changing interests. The reason is simple; in opportunistic network, stable connections cannot be guaranteed, nor a centric security service.

(2) *Construction of Forwarding Tables*. If one subscriber A_i wants to broadcast its interest advertisement, it chooses the necessary trapdoor sets from all candidate ones obtained when registering, then uses the trapdoor sets as new CI_{A_i}, and broadcasts new RA_i to all intermediate nodes it meets. After intermediate node B_k receives RA_i, it will create a new record in its forwarding table composed of two parts; the first part is CI_{A_i} of RA_i and the second is the identity of A_i. The process is described in Figure 4.

When intermediate B meets D, it firstly does the lookup of FT_B and delivers messages in $List_B$. Then it fetches all records from its FT_B, adds self-identity to the second part of every forwarding table record, and exchanges it with D.

(3) *Secure Lookup of Forwarding Tables*. When publisher C_j has new payload P_{C_j} to publish, it chooses keywords that can index the payload to form a set $W_{C_j} = \{w_{C_j,1}, w_{C_j,2}, \ldots, w_{C_j,M_{C_j}}\}$ and then encrypts them based on *PEKS* to get a set of searchable encrypted keywords which will be used as new CI_{C_j}:

$$PEKS_{C_j} = \left\{w' = PEKS\left(TTP_{\text{pub}}, w\right), \text{ for each } w \text{ in } W_{C_j}\right\}. \quad (10)$$

With CI_{C_j}, the encrypted payload $E(P_{C_j})$ and the access policy $T_{P_{C_j}}$ together, C_j forms a new PC_j and broadcasts it to every intermediate node it meets. Details about $E(P_{C_j})$, and $T_{P_{C_j}}$ will be discussed in the following section. The process is described in Figure 5.

When intermediate node B meets some other one D, it would execute *SecureLookup* algorithm and decide whether to forward some messages to it.

In *SecureLookup* algorithm, the subfunction *FuzzyTest* function is used to calculate the similarity between S_{TD} and $S_{w'}$ (see Algorithms 1 and 2). Here S_{TD} means the control information from a record of forwarding table and $S_{w'}$ means the control information of a published content. A larger returned value means a higher similarity. If the result of *FuzzyTest* is higher than threshold, the payload will be delivered to D.

```
Foreach ft in FT_B
  If (D ∈ ft · SID) == true)
    Foreach pc in List_B
      similarity = FuzzyTEST (ft · RI, pc · CI);
      If (simlarity ≥ TH) {TH: threshold to deliver}
          deliver pc · P to D;
          break;
      End
    End
  End
End
```

Algorithm 1: *SecureLookup* (FT_B, D).

```
Define Counter = 0
Foreach td in S_TD
  Foreach w' in S_w'
    If (Test (w', td) == true) Counter ++;
    End
  End
End
Return Counter;
```

Algorithm 2: *FuzzyTest* (S_{TD}, $S_{w'}$).

4.3. CP-ABE-Based Confidentiality Preserving Scheme

(1) *Initialization*. TTP is responsible for initialization of *CP-ABE* including the creation of public key CPA_{pub} and the master key CPA_{master}. Having finished the key generation,

TTP will broadcast CPA_{pub} to every user intending to participate in the system. The registration stage also involves authentication which is beyond the scope of this paper. After the registration, a subscriber A_i submits its attributes set Y_{A_i} to the trusted third party which is responsible for calculating the corresponding private key $CPA_{\text{priv},Y_{A_i}}$ and returning it to A_i. After the registration, *TTP* could be away from the connected areas of the system:

$$CPA_{\text{priv},Y_{A_i}} = CPA\_KeyGen\left(CPA_{\text{master}}, CPA_{\text{pub}}, Y_{A_i}\right). \tag{11}$$

(2) *Encryption of Published Content*. When a publisher C_j has new contents to publish, he firstly creates an access policy in which designated subscribers' attribute requirements are described, and then he would encrypt the content with CPA_{pub} to get cipher text $E(P_{C_j})$:

$$CPA_{\text{priv},Y_{A_i}} = CPA\_KeyGen\left(CPA_{\text{master}}, CPA_{\text{pub}}, Y_{A_i}\right). \tag{12}$$

(3) *Decryption of Published Content*. The encrypted published content message is then forwarded according to some kind of content-based routing scheme which of course could be our proposed *PEFKS*-based interest-centric routing scheme. When a subscriber who is interested in the content receives the message, it tries to decrypt. If and only if the subscriber's personal attribute set satisfies the access policy, the decryption succeeds and he gets the plaintext:

$$P_{C_j} = CPA\_Decrypt\left(CT, CPA_{\text{priv},Y_{A_i}}, E\left(P_{C_j}\right)\right). \tag{13}$$

The subscheme mentioned previously is illustrated in Figure 6.

5. Evaluation and Analysis

In this chapter we will evaluate and analyze the security and the performance of our schemes.

5.1. Security. First is the security requirement of privacy. In our scheme, users' full privacy is well preserved by *PEFKS*.

PEFKS is reliable in terms of cryptography concerns, owing to deriving from *PEKS*. In [9], Boneh proved that PEKS is able to resist chosen keyword attack in random oracle model [21] in the assumption that bilinear Diffie-Hellman problem is difficult. In other words, having no trapdoors of keywords, obtaining the plain text of keyword is impossible. In our *PEFKS* utilizing, the creation process of trapdoors relies on private key of *TTP* which indicates that only *TTP* is allowed to calculate and distribute the trapdoors. To support search on encrypted fuzzy keywords, a fuzzy set related to every submitted keyword is created firstly. For every single word in a fuzzy set, the analysis is the same as in the case of PEKS [9].

Some people might have doubts about the possibility of brute force attack performed by adversary who has almost unlimited calculation power and storage space. The adversary would try generating as many encrypted keywords as possible, using the public key. In our proposed solution, fuzzy interests add a second layer of anonymity for users by blurring users' real interests. The reason is that only one keyword in a fuzzy set could reveal real interests of some users. Even if an adversary consumes huge time and space to achieve the indexes of published contents, what he actually gets is only a set of fuzzy words, which makes it very hard to acknowledge user's true privacy. Such a trick corresponds to k-anonymity protection model introduced in [22].

As for threats caused by malicious users who are already authenticated by *TTP*, they also could not work out in our proposed scheme. The most dangerous security pitfall is

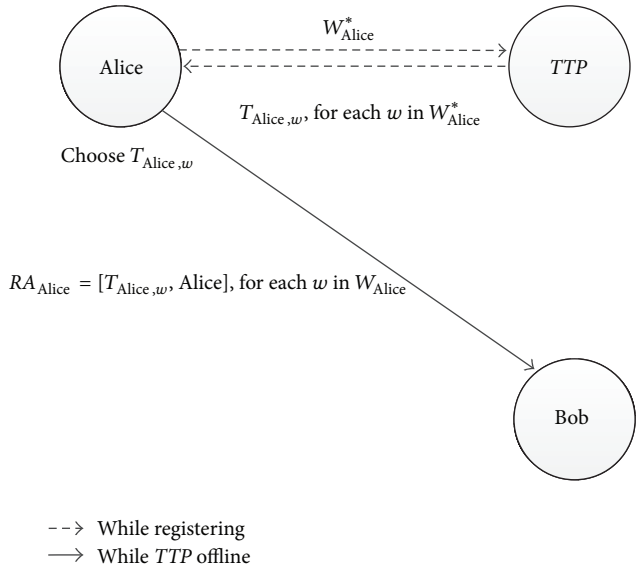


Figure 4: Setup of one record in forwarding table.

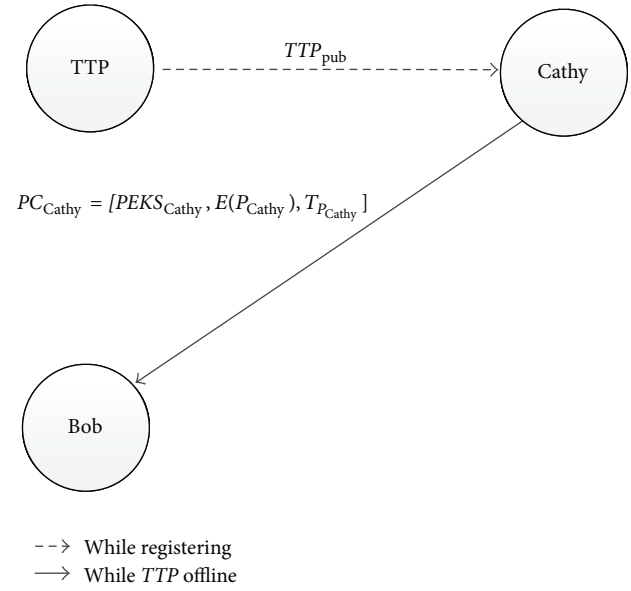


Figure 5: Create and publish new content.

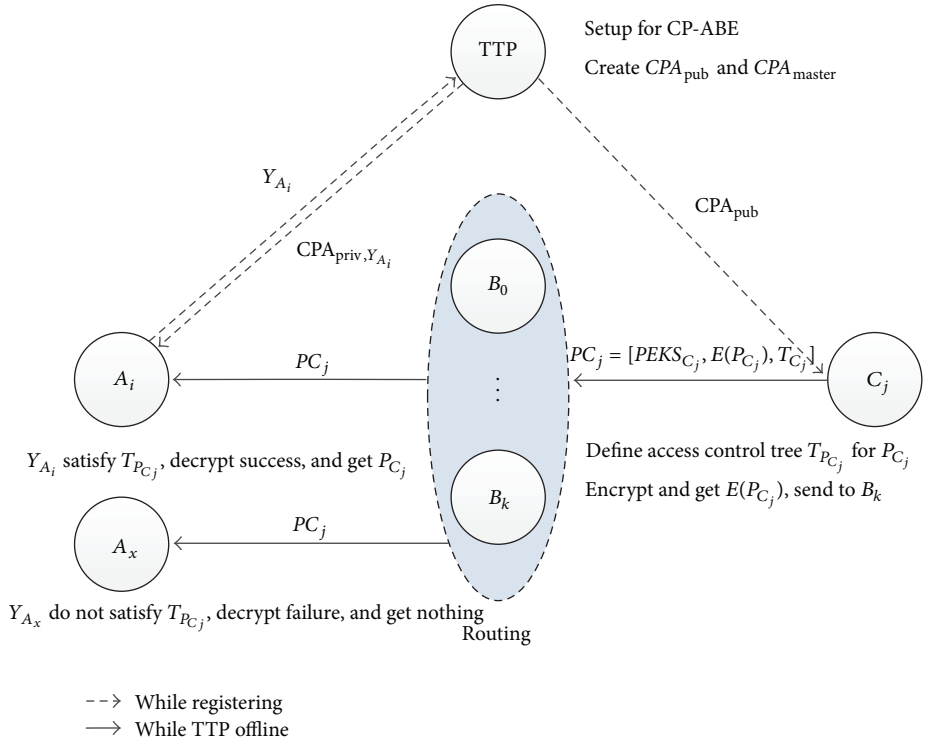


FIGURE 6: *CP-ABE*-based confidentiality protecting scheme.

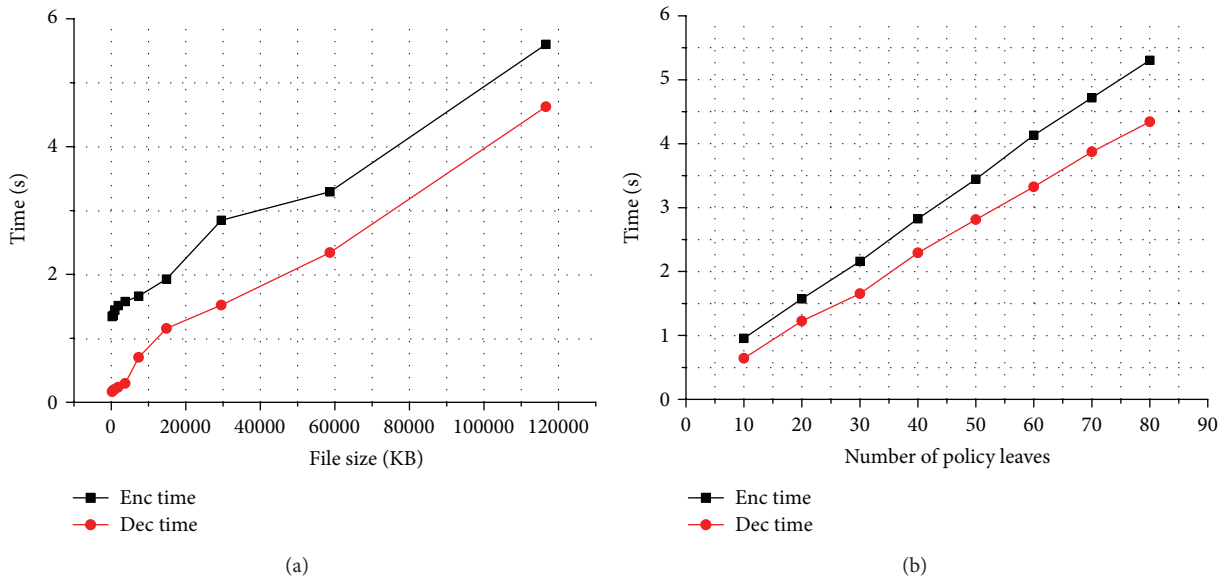


FIGURE 7: *CP-ABE* encryption and decryption time cost: (a) the x-axis corresponds to the complexity of the access policy; (b) the x-axis corresponds to the complexity of access control strategy tree.

when malicious users act as intermediate nodes, which is prevented completely in our scheme. On one hand, creation and exchange of forwarding tables involve no cryptographic operations on encrypted keywords which keeps valuable information safe. On the other hand, lookup of forwarding tables and decisions of forwarding is based on *FuzzyTest*, in which the similarity is calculated using a trapdoor set and a searchable encrypted keyword set as inputs. In all these processes, neither plaintext of interests nor relation between an interest keyword and a user is leaked. Thus full privacy presented by Shikfa et al. [1] is achieved.

Second is the confidentiality requirement. In loosely trusted environment of opportunistic network, the main challenge for confidentiality is collusion attack, which is also

Table 1: Comparison of security features.

Security feature	Nguyen et al. [4]	Symington et al. [5]	Shikfa et al. 2009 [6]	Shikfa et al. 2010 [1]	Proposed scheme
End-to-end payload encryption	Yes	Yes	Yes	Yes	Yes
Resilience to dictionary attack	No	Yes	Yes	Yes	Yes
User's full privacy	No	No	Yes	Yes	Yes
Decryption based on content/context	No	No	No	Yes	Yes
Secure partial match in forwarding	No	No	No	Yes	Yes
Secure fuzzy keyword index	No	No	No	No	Yes
Secure multiple keyword index	No	No	No	No	Yes
Secure access control strategy for message	No	No	No	No	Yes

Table 2: File size before and after encryption of *CP-ABE*.

Published content ID	File size (KB)		
	Before encryption	After encryption	Added size
1	239	241	2
2	473	475	2
3	951	953	2
4	1843	1845	2
5	3711	3713	2
6	7365	7367	2
7	14844	14846	2
8	29511	29513	2
9	58678	58680	2
10	116632	116634	2

the core challenge of *ABE*. In *CP-ABE*, *KeyGen* uses two layers of random masking to create private keys. Private key for every user is related to the second layer of random masking; thus even if two or more subscribers conclude sharing their submitted attributes and related private keys, the collusion attack will not take effect. The detailed demonstration is illustrated in [16, 19].

Comparison of security features between our proposed scheme and other aforementioned schemes in context of content-based opportunistic network is shown in Table 1. It could be seen that the proposed scheme achieves most security features of them.

5.2. Performance. The performance of *PEFKS* mainly relies on the size of the trapdoor set from a forwarding table record and the size of the searchable keyword set from a published content. Dictionary-based filter strategy used here could decrease the size of a fuzzy set dramatically, thus speeding up the calculation of *FuzzyTest* algorithm.

To evaluate the performance of *CP-ABE*, we conducted several experiments on a virtual machine equipped with Core 2 Duo based on jPBC library [23] which is a java version PBC library [24]. We measured the time required for encryption and decryption under various scenarios. Besides, we measured the cipher text size overhead incurred by our scheme to see if an acceptable cost exists in storage.

From Figure 7 we can see that the encryption and decryption time of *CP-ABE* has a significant linear correlation with the size of published content and the complexity of access policy. Considering the file size, even for a file of 120 Mbytes, it costs less than 6 seconds. Considering the complexity of access policy, encryption and decryption time for an access control strategy tree with 80 leaves is still no more than 6 seconds. In an age that mobile devices strong in computing power and storage are widespread, such a time cost is acceptable. What is more, in the application environment of opportunistic network, publishers or subscribers are not worried to encrypt or decrypt, which will leave enough time for security operations.

From Table 2. we can see that no matter what size the original published content is, the cipher text is always 2 Kbytes larger, meaning *CP-ABE* costs almost no storage redundancy in practice.

6. Conclusion

In this paper, we introduced the concept of interest-centric and attribute-based identities into opportunistic network. We focused on the security issues of user privacy and end-to-end confidentiality in a specific distributed application scenario of wireless opportunistic network, in which people work in near areas and share information without revealing their identity.

Finally, we proposed our *PEFKS*- and *CP-ABE*-based distributed security scheme which consists of two sub schemes. The first is a *PEFKS*-based privacy preserving forwarding decision scheme which assures users' full privacy and enhances anonymity through fuzzy interests. As far as we know, we are the first to employ fuzzy technique in opportunistic network and to allow partial match on encrypted fuzzy control information. The second is a *CP-ABE*-based confidentiality protecting scheme, in which publishers make and attach an expressive access control strategy to the messages they are about to send, and then subscribers whose attribute-based identities satisfy the access control strategy are legalized to decrypt cipher-text, finally achieving confidentiality with a fine-grained access control strategy on shared data. Both schemes rely on an offline *TTP* which is needed only when opportunistic network user registers.

We evaluated our scheme from aspects of security and performance, which leads to the conclusion that our scheme

suits the features of opportunistic network very well, because of not only the relatively low cost in computation and storage but also its satisfaction with necessitous need for fine-grained access control.

Acknowledgments

This paper is supported in part by Important National Science & Technology Specific Projects under Grant nos. (2010ZX03006-002, 2010ZX03006-007), National Basic Research Program of China (973 Program) (no. 2011CB302803), and National Natural Science Foundation of China (NSFC) under Grant no. (61173132, 61003307). The authors alone are responsible for the content of the paper.

References

[1] A. Shikfa, M. Önen, and R. Molva, "Privacy and confidentiality in context-based and epidemic forwarding," *Computer Communications*, vol. 33, no. 13, pp. 1493–1504, 2010.

[2] P. Hui, A. Chaintreau, R. Gass, J. Scott, J. Crowcroft, and C. Diot, "Pocket switched networking: challenges, feasibility and implementation issues," *Lecture Notes in Computer Science*, vol. 3854, pp. 1–12, 2006.

[3] L. Lilien, Z. Kamal, V. Bhuse, and A. Gupta, "Opportunistic networks: the concept and research challenges in privacy and security," in *Proceedings of the NSF International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks*, 2006.

[4] H. A. Nguyen, S. Giordano, and A. Puiatti, "Probabilistic routing protocol for intermittently connected mobile ad hoc network∗ (PROPICMAN)," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '07)*, pp. 1–6, Espoo, Finland, June 2007.

[5] S. Symington, S. Farrell, H. Weiss, and P. Lovel, *Bundle Security Protocol Specification Draft-Irtf-Dtnrg-Bundle-Security-19*, 2011.

[6] A. Shikfa, M. Önen, and R. Molva, "Privacy in content-based opportunistic networks," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA '09)*, pp. 832–837, Bradford, UK, May 2009.

[7] N. Li and S. K. Das, "A trust-based framework for data forwarding in opportunistic networks," *Ad Hoc Networks*, 2011.

[8] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "AnonySense: a system for anonymous opportunistic sensing," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 16–30, 2011.

[9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *Lecture Notes in Computer Science*, vol. 3027, pp. 506–522, 2004.

[10] S. Ji, G. Li, C. Li, and J. Feng, "Efficient interactive fuzzy keyword search," in *Proceedings of the 18th International Conference on World Wide Web*, Madrid, Spain, 2009.

[11] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proceedings of the IEEE (INFOCOM '10)*, pp. 1–5, San Diego, Calif, USA, March 2010.

[12] E. S. Ristad and P. N. yianilos, "Learning string-edit distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

[13] L. Chang, Z. Liehuang, L. Longyijia, and T. Yuran, "Fuzzy keyword search on encrypted cloud storage data with small index," in *Proceedings of the IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS '11)*, pp. 269–273, 2011.

[14] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. Blakley and D. Chaum, Eds., vol. 196, pp. 47–53, Springer, Berlin, Germany, 1985.

[15] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of the Advances in Cryptology (CRYPTO'01)*, J. Kilian, Ed., vol. 2139, pp. 213–229, Springer, Berlin, Germany, 2001.

[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT '05)*, pp. 457–473, May 2005.

[17] J. S. Su, D. Cao, X. F. Wang, Y. P. Sun, and Q. L. Hu, "Attribute-based encryption schemes," *Journal of Software*, vol. 22, pp. 1299–1315, 2011.

[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, Alexandria, VA, USA, November 2006.

[19] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the S and P IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, Berkeley, Calif, USA, May 2007.

[20] A. Shikfa, M. Onen, and R. Molva, "Privacy-preserving content-based publish/subscribe networks," *Emerging Challenges for Security, Privacy and Trust*, vol. 297, pp. 270–282, 2009.

[21] X. Jia, B. Li, and Y. Liu, "Random oracle model," *Journal of Software*, vol. 23, pp. 140–151, 2012.

[22] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[23] jPBC, "Java Pairing Based Cryptography," http://gas.dia.unisa.it/projects/jpbc.

[24] PBC, "Paring Based Cryptography," http://crypto.stanford.edu/pbc.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 812438, 8 pages
http://dx.doi.org/10.1155/2013/812438

Research Article

OPCPP: An Online Plug-Configure-Play Experiment Platform for WSN

Wei Lin,[1] XueHai Hong,[1] Chaonong Xu,[2] YongJun Xu,[1] and DengChang Zhao[1]

[1] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[2] *Department of Computer Science and Technology, China University of Petroleum, Beijing 102249, China*

Correspondence should be addressed to XueHai Hong; hxh@ict.ac.cn

Received 28 December 2012; Accepted 11 March 2013

Academic Editor: Hongsong Zhu

Wireless sensor network (WSN) experiment platforms are used for teaching, research, and development of WSN. However, existing WSN experiment platforms generally have the following disadvantages: tedious manual operations, invasive measurement method, poor sensor self-awareness, and low resource utilization rate. To address the above problems, this paper proposes OPCPP, an online Plug-Configure-Play experiment platform for WSN. It has four prominent strongpoints: in-application programming in batch, noninvasive measurement method, sensor self-awareness, and remote operation. OPCPP has been used in eight colleges in China for teaching course of "internet of things" till now. We also develop a sensor-aware ZigBee-based smart home system prototype based on OPCPP.

1. Introduction

Recently, wireless sensor network has a great research progress because of the advances in microelectromechanical systems and wireless communication technologies. To have a deep understanding and application of WSN, WSN experiment platform is essential for development, research, and teaching of WSN. However, the existing WSN experiment platforms seldom have good usability and are mainly limited in the following aspects.

(1) Tedious Manual Operations. WSN experiments often involve multiple nodes. Therefore, when users conduct an experiment on current experiment platform, they often spend a lot of time in repeating tedious operations for every node, which results in low efficiency. For example, for having an experiment of wireless routing protocol, users have to download object code into nodes in WSN experiment platform one by one with a hardware emulator, such as CC Debugger in TI [1]. Obviously this inevitably involves repeated manual plug-and-program operations which are time consuming and tedious.

(2) Invasive Measurement Method. In existing experiment platforms, the traditional method for collecting experimental data is as follows: node transfers its experimental data to the sink node via wireless communication, and then the sink node delivers all data to PC through serial port or ethernet port for further process. Since a large number of experimental data collected via wireless channel could fail the wireless network, this method is obviously not suitable for experiments such as time synchronization experiment [2], routing algorithm experiment [3], and other experiments where a large number of experimental data has to be collected for analyzing protocol performance.

(3) Poor Sensor Self-Awareness. The existing experiment platforms generally support the experiment of several common sensors, such as temperature, humidity, and accelerometer. Although the driver development of these sensors is similar,

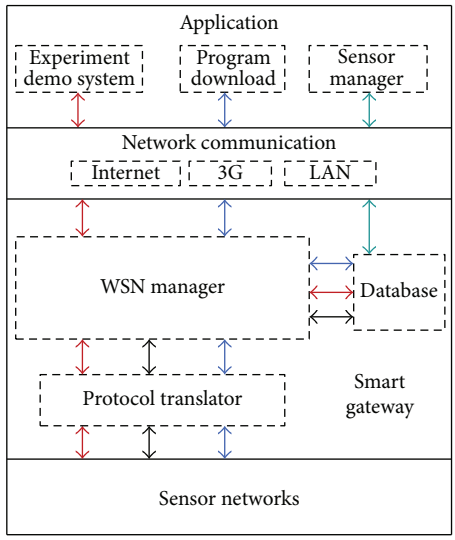


Figure 1: Architecture of OPCPP.

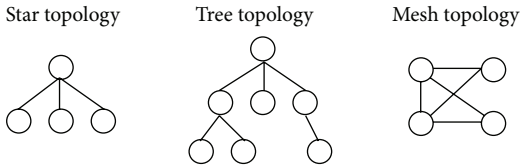


Figure 2: Sensor network topology supported.

it is still difficult to a beginner. An easy-to-use experiment platform should have the ability of self-adaption for new sensor.

(4) Low Resource Utilization Rate. Existing platforms generally do not support remote operation. Users can use the experiment equipment only when they are in the lab. Thus, low resource utilization rate is inevitable.

In order to address the above-mentioned problems, we design OPCPP, an online Plug-Configure-Play WSN experiment platform. Contrary to the above four shortcomings of traditional WSN experiment platforms, the new experiment platform has four strongpoints as follows.

(1) In-Application Programming in Batch. Users can update the code of any node in OPCPP without hardware emulator, and they can even update codes of all nodes in batch simultaneously with our software tool. This will significantly reduce the tedious and repeated manual code downloading operations.

(2) Noninvasive Measurement Method. Two data channels, wireline or wireless, are built in OPCPP, and users can assign either of the two channels for data transmission. A typical application scenario is that the test data is transmitted via wireline while the data of protocol execution process, such as Transport Protocol Data Unit (TPDU), is transmitted wirelessly. With this assignment, the test data flows do not interfere with TPDU, so the performance of wireless protocol

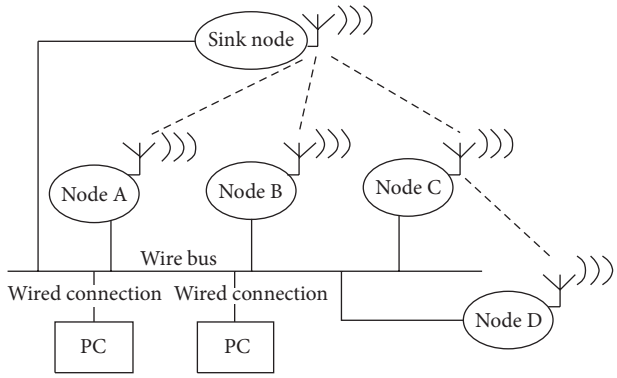


Figure 3: Architecture of sensor network layer.

can be accurately measured by analyzing test data post facto. The feature is especially important for research in wireless network.

(3) Sensor Self-Awareness. The concept of Plug-Configure-Play is similar to the concept of Plug-and-Play. It means that when a node plugs a new sensor, user only needs to specify the usage method of the sensor with simple operation on configuration software running on PC, and the sensor can then work without any driver developing. Thanks to this capability, it seems that node has the capability of sensor self-awareness only if the electric interface of sensor is compatible with node.

(4) Remote Operation. OPCPP can be connected to a local network or Internet. Users can remotely access any node in OPCPP and get any data they need with a browser such as IE. Users can also download program of nodes remotely. It is convenient for user to conduct an experiment whenever and wherever. It will enhance the resource utilization rate significantly and improve the availability of OPCPP.

The remainder of this paper is organized as follows. In Section 2 related works are introduced. In Section 3, architecture of OPCPP is presented. Section 4 gives the implementation of the four strongpoints of OPCPP. OPCPP is shown in Section 5, and the last section is conclusions and future works.

2. Related Works

In the section, we show briefly the existing WSN experiment platforms. Many platforms have been designed for one or several specific emphasis. Some existing platforms focus on teaching of WSN. Ying-xin et al. [4] researched and implemented a teaching experiment platform based on MSP430 and CC2420 chipsets. This platform is mainly a simple temperature and humidity data acquisition and presentation system for teaching WSN. Sun and Andrew Yang [5] presented a suite of practical WSN applications based on TinyOS. It focuses on using WSN data collection and dissemination as examples to provide technology-based education and training. An-Feng et al. [6] proposed WSN routing experimental teaching platform based on Omnet++. It focuses on the

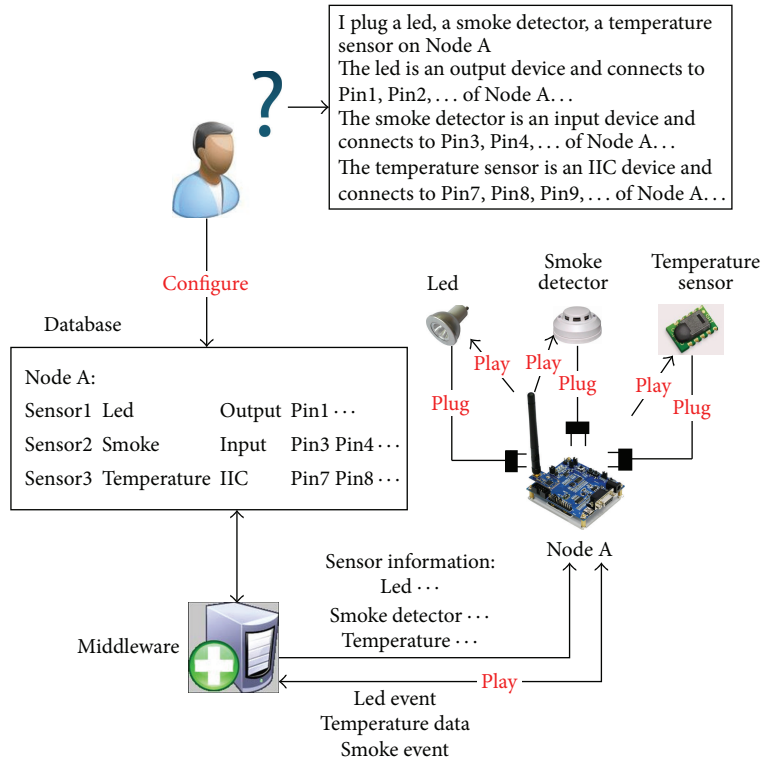


Figure 4: Sensor self-awareness and Plug-Configure-Play.

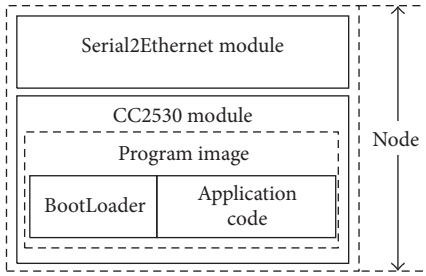


Figure 5: Structure of sensor node.

teaching and experiment of routing algorithm. Hloupisa et al. [7] proposed an open source development platform for WSN, which is designed for simple WSN teaching.

Some existing platforms focus on providing development platform of WSN. Girod et al. [8] proposed Emstar: a software environment for developing and deploying heterogeneous sensor-actuator networks. It is a software environment for developing and deploying complex WSN applications. Kuo et al. [9] presented an NTP experimental platform for heterogeneous wireless sensor networks. They mainly considered providing an open platform to test and to showcase WSN systems and applications. Handziski et al. [10] proposed a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. It is mainly designed for indoor deployment of wireless sensor networks. Ertin et al. [11] proposed a testbed for sensing at scale; its design focuses on sensing and scaling. Gireesan Namboothiri et al. [12] designed a testbed for distributed target tracking with directional sensors.

Some platforms mainly consider WSN research. Schiller et al. [13] proposed ScatterWeb, which is a wireless sensor net platform for research and teaching. This platform focuses on the design for low-power consumption. Werner-Allen et al. [14] proposed Motelab: a web-based wireless sensor network testbed. It is a powerful tool mainly for sensor network research. Kuzu et al. [15] introduced a multipurpose WSN platform for research and training in data fusion and multifeature target tracking. Sheu et al. [16] designed and implemented a testbed to realize various experiments in heterogeneous wireless sensor networks.

Compared to the existing related research, we focus on the experiment efficiency, availability, and scalability of WSN experiment platform.

3. Architecture of OPCPP

In this section, we introduce the architecture of OPCPP. In order to improve flexibility and scalability of the platform, we design a layered architecture. As shown in Figure 1, the architecture of OPCPP consists of four layers: sensor network layer, smart gateway layer, network communication layer, and application layer.

3.1. Sensor Networks Layer. This layer consists of a number of nodes which carry sensors. As shown in Figure 2, the nodes can compose wireless networks of different topology which

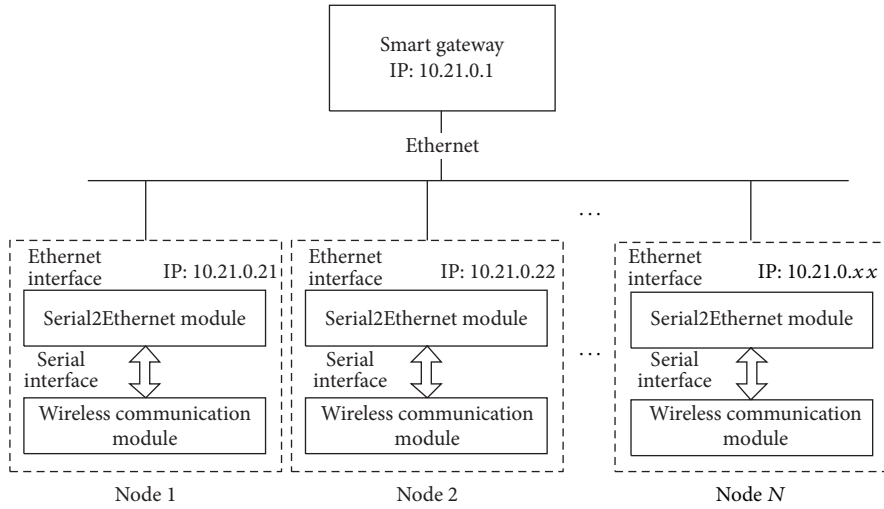


Figure 6: Providing wireline channel for noninvasive measurement.

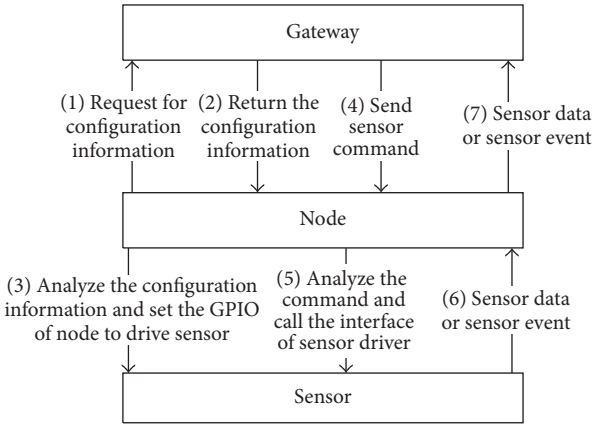


Figure 7: Process of "play" in Plug-Configure-Play.

are important for collecting the sensor data and controlling the sensor action.

As mentioned above, in order to provide the capability of invasive measurement method, each node of OPCPP has both wireless communication channel and wireline communication channel, as shown in Figure 3. Node can collect sensor data and transmit sensor data to sink node in wireless channel while background data such as the status of node and test data can be transmitted via wireline channel.

3.2. Smart Gateway Layer. This layer is mainly a Plug-Configure-Play service-oriented gateway which is proposed by our research partners [17]. The smart gateway layer consists of WSN manager, protocol translator, and database. WSN manager supervises the sensor network and provides simple application interfaces. Protocol translator executes the protocol conversion between ZigBee and IP-based protocol for convenient access from external network. Database stores the information of sensors on the nodes, test data, and association rules, which is a necessary component for sensor self-awareness.

A creativeness in this layer is the so-called "Plug-Configure-Play." As shown in Figure 4, when user plugs a sensor or an actuator, such as a LED, a smoke detector, or a temperature sensor on the node, user configures the usage information of sensors, such as the output pin, input pin, and enable level, in the first step, then this information will be saved in the database. WSN manager sends the above information to node carrying the sensor. The node can then play the device based on the information. In this manner, new sensor can be supported without driver development.

3.3. Network Communication Layer. This layer provides multiple ways for application layer to access gateway layer, such as Internet, 3G, Wi-Fi, and LAN.

3.4. Application Layer. The application layer includes three components: experiment demo system, program download module, and sensor manager. Experiment demo system dynamically shows collected sensor data and controls the actions of sensor. This system is suitable for teaching users to understand the concept of WSN and conduct experiments. Program download module provides an interface to download programs to nodes via local network or Internet. This will significantly reduce the repeated manual operations and improve the experiment efficiency. Sensor manager is used for configuring the information of sensors on nodes in order to conduct different sensor experiments flexibly.

4. The Implementation of OPCPP

In this section, we give the implementation of OPCPP by introducing our solution to the main creativeness of OPCPP.

4.1. In-Application Programming. The node of OPCPP consists of a CC2530 module and a Serial2Ethernet module as

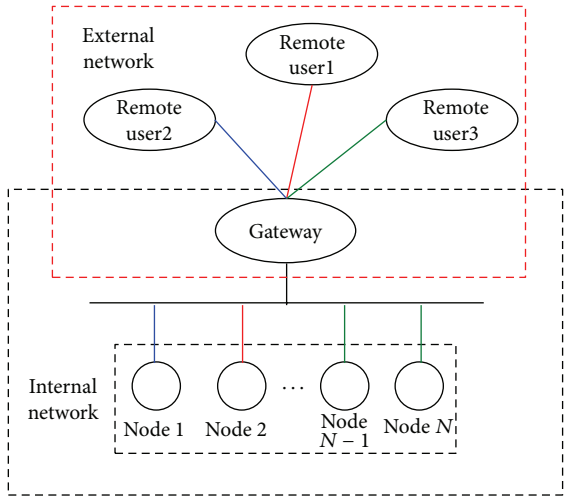


Figure 8: Architecture of remote operation.

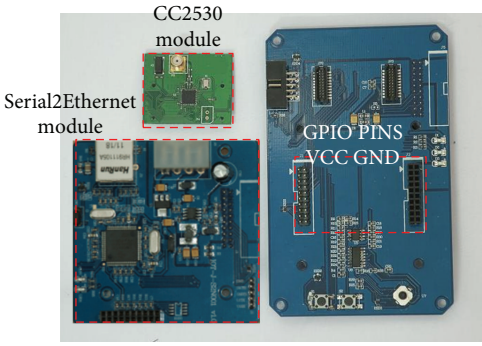


Figure 9: Prototype of Node.

shown in Figure 5. CC2530 [18] is a system-on-chip (SoC) solution for IEEE 802.15.4, ZigBee, and RF4CE applications. It enables robust network nodes to be built with very low total bill-of-material costs. The CC2530 combines the excellent performance of a leading RF transceiver with an industry-standard enhanced 8051 MCU, in-system programmable flash memory, 8-KB RAM, and many other powerful features.

In order to implement the function of in-application programming, the program image of CC2530 is divided into two sections. As shown in Figure 5, the first section is for BootLoader and the second section for application code which stores user program. BootLoader is used for reprogramming the application code section. BootLoader receives the packets of application code which users want to download from serial port (serial port can be remotely controlled via the under-mentioned Serial2Ethernet module) and then writes them into the specified flash area in the application code section. After downloading the application code, BootLoader will jump to the application code section for running the application code.

4.2. Noninvasive Measurement Method. Node provides wire-line channel to support noninvasive measurement method. We use a Serial2Ethernet module as shown in Figure 6, the Serial2Ethernet module realizes conversion between serial communication and ethernet communication. Each Serial2Ethernet module has a configured IP address which can be used for identifying nodes. With ethernet cable, each Serail2Ethernet module connects node to smart gateway with ethernet. So, user can do experiment of WSN and acquire test data without influencing the execution of wireless protocol.

4.3. Sensor Self-Awareness. Critical idea for sensor self-awareness is to migrate sensor driver from node in sensor network layer to smart gateway layer. When a new sensor is plugged on a physical node, user will configure the sensor in the user interface of smart gateway layer. The information includes the type of a sensor, sensor name, the GPIO pins of node connected with the sensor, and so forth. Then the database in smart gateway will store the information.

As shown in Figure 7, the process of "play" in Plug-Configure-Play is as follows: (1) when a node starts to work, it will request smart gateway to send its configuration information (the information of virtual nodes on the node). (2) Smart gateway sends the configuration information of the request node. (3) After the node receives the configuration information packet, it will analyze the packet based on the protocol defined beforehand. The configuration packet includes the type of sensors, virtual address, the IO pins which sensors connect to, and so forth. The node sets its GPIO pins and records the information of sensors which it connects to. Then sensors on the node can be driven normally. (4) When application layer calls service interface to control a sensor, gateway sends commands to the node which the sensor belongs to. (5) The node receives the command and calls the interface of sensor driver. (6) Sensor returns data to the node or operates according to the command. (7) The node returns the sensor data or sensor event to gateway.

4.4. Remote Operation. To support remote operation of OPCPP, it should be connected to a local network or Internet.

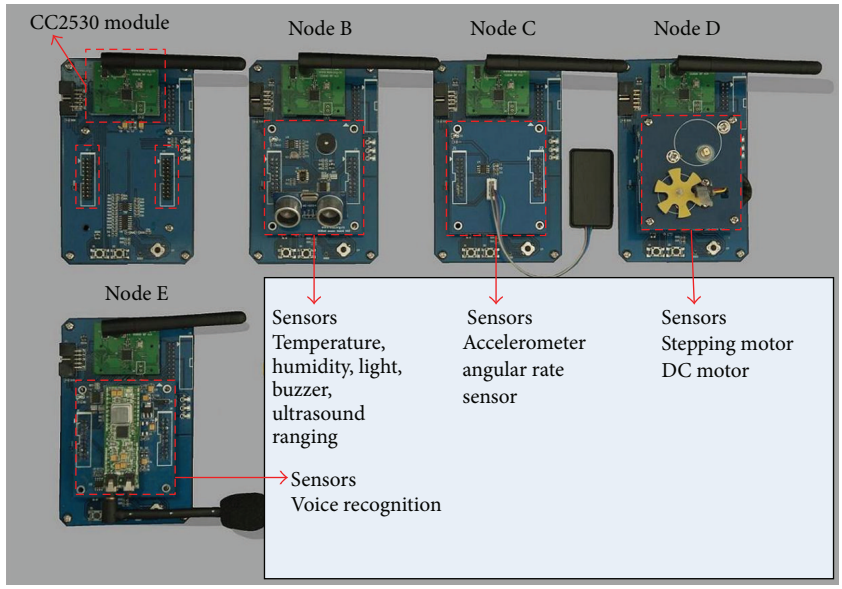


Figure 10: OPCPP overview.

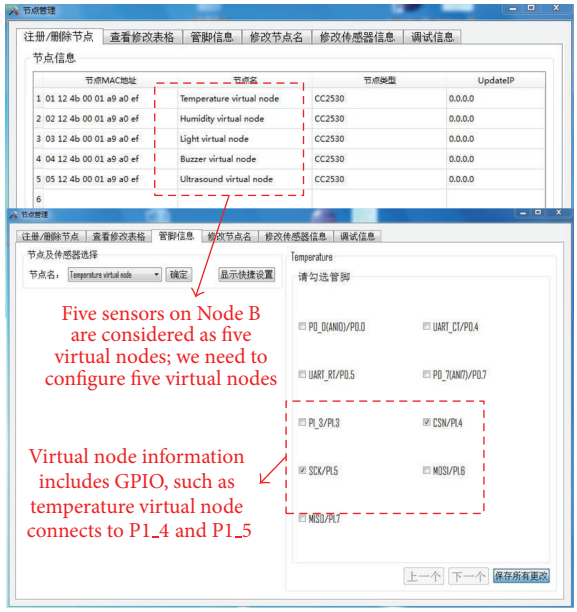


Figure 11: Configuration of sensor information.

As above mentioned, every node has a Serial2Ethernet module. Every Serial2Ethernet module implements a lightweight TCP/IP protocol stack. Thus, smart gateway and users can establish a TCP/IP connection to every node. With the TCP/IP connection, smart gateway and users can remotely access every node, update program, and get data of every node. As shown in Figure 8.

5. OPCPP Show

The prototype of OPCPP is shown in Figure 9 as follows. Each node consists of a CC2530 module and Serial2Ethernet module. Each node has 21 GPIO pins, VCC pin, and GND pin. The pins are used for connecting to sensors.

As shown in Figure 10, various sensors are plugged on each node. For example, Node B has five sensors: temperature, humidity, light, buzzer, and ultrasound ranging. Node C has two sensors: accelerometer and angular rate sensor.

When sensors are plugged on a node, we configure the sensor information of the node as shown in Figure 11. The information includes node address, node name, and GPIO pins used by sensors.

After configured, the sensors can work. Node will collect the data of sensors and transmit data to gateway, and then experiment demo system shows these sensor data in graphic manner. For example, the four sensors including temperature, humidity, light strength, and ultrasound ranging are on Node B. After configured as shown in Figure 11, the four sensors on Node B can work. The data of the four sensors will be collected and shown in the experiment demo system. Figure 12 is graphic interface of the four sensor data. Users can watch the real-time changes of the four sensor data.

OPCPP provides an online program download interface for users' development and research of WSN. Users can also easily establish a TCP/IP connection to the experiment node and get test data using popular software such as network debugger assistant. For example, as shown in Figure 13, we download the code to all the nodes in OPCPP simultaneously. The IP of Node 2 and Node 3 are, respectively, 192.168.1.22 and 192.168.1.23. After programming, we establish a TCP/IP connection to Node 2 and Node 3 with a network assistant and get the test data of Node 2 and Node 3.

After our experiment statistics, the average time of programming a node using a hardware emulator is about 2.3 minutes. The average time of programming a node in OPCPP is about 1.4 minutes. The average time of programming all nodes (10 nodes) in OPCPP simultaneously is about 1.45 minutes. The time to complete an experiment using a

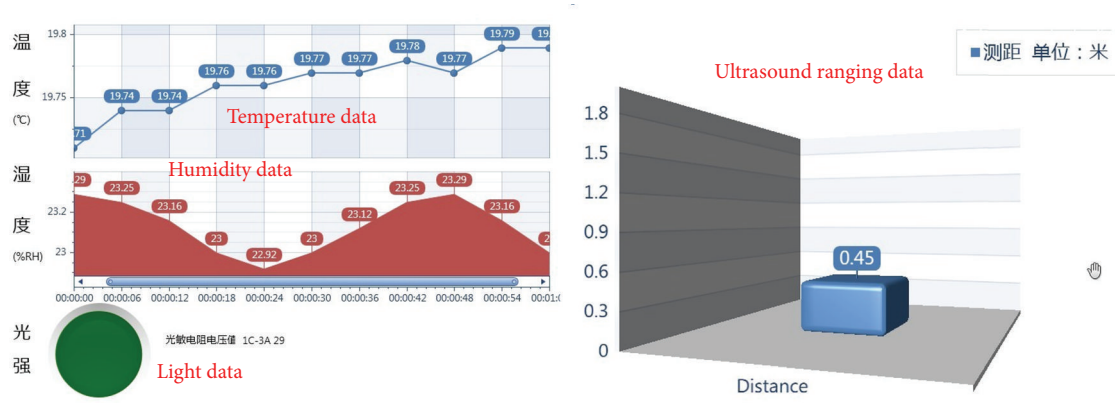


Figure 12: Experiment demo.

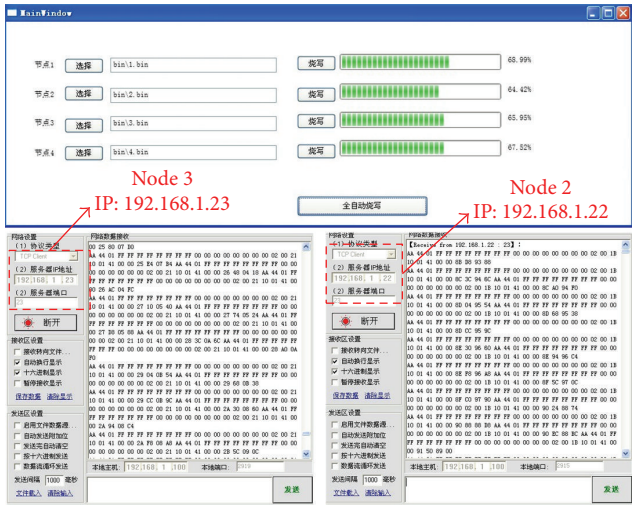


Figure 13: Download code and collect test data with network debugger assistant.

hardware emulator is proportional to the number of nodes it needs. However, the time to complete an experiment using OPCPP is almost a constant because the process speed of PC is far faster than the nodes and the programming of nodes can be concurrent. Thus, it will greatly reduce the experiment time especially when the WSN experiment involves many nodes.

6. Conclusions and Future Works

Current experiment platform cannot satisfy the need of teaching, research, and development of WSN. In order to provide a simple, flexible, and scalable platform, this paper proposes OPCPP, an online Plug-Configure-Play experiment platform for WSN. It has four prominent strongpoints: in-application programming in batch, noninvasive measurement method, sensor self-awareness, and remote operation. With OPCPP, users can conveniently do application development and research for algorithms and protocols in WSN. It also improves the utilization rate of hardware resources. We also quickly developed a ZigBee-based smart home system prototype based on OPCPP.

For future works, we will improve OPCPP to provide more simple interfaces. We will attempt to combine it with cloud computing technology to build a virtual WSN lab for users. We will also research for algorithms and protocols of wireless network with OPCPP.

Acknowledgment

This work was supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China under the Contract no. 2011ZX03005-005.

References

[1] http://www.ti.com/lit/ug/swru197e/swru197e.pdf.

[2] Z. An, H. Zhu, X. Li, C. Xu, Y. Xu, and X. Li, "Nonidentical linear pulse-coupled oscillators model with application to time synchronization in wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2205–2215, 2011.

[3] J. Zhang and L. Gao, "Clustering routing algorithm for WSN based on improved ant colony algorithm," in *Proceedings of the 2nd Annual Conference on Electrical and Control Engineering (ICECE '11)*, pp. 2924–2928, Yichang, China, 2011.

[4] Z. Ying-xin, X. Ping-ping, X. Gui-bin, and S. Tie-cheng, "Design and realization of teaching experiment based on wireless sensor networks," *Journal of Electrical & Electronic Education*, no. 5, pp. 48–53, 2008.

[5] B. Sun and T. Andrew Yang, "WSNED: a suite of data collection and dissemination applications for Wireless Sensor Network Education," *Journal of Computing Sciences in Colleges*, vol. 25, no. 4, pp. 104–111, 2010.

[6] L. An-Feng, R. Ju, and C. Zhi-Gang, "WSN routing experimental teaching platform based on omnet++," *Computer Engineering*, vol. 38, no. 11, pp. 258–261, 2012.

[7] G. Hloupisa, I. Stavrakas, K. Moutzouris, A. Alexandridis, and D. Triantis, "WSN open source development platform: application to green learning," *Procedia Engineering*, vol. 25, pp. 1049–1052, 2011.

[8] L. Girod, N. Ramanathan, J. Elson, T. Stathopoulos, M. Lukac, and D. Estrin, "Emstar: a software environment for developing and deploying heterogeneous sensor-actuator networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 3, article 13, Article ID 1267061, 2007.

[9] S.-P. Kuo, C.-Y. Lin, Y.-F. Lee, and W. Hua- Fang, "The NTP experimental platform for heterogeneous wireless sensor networks," in *Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, 2008.

[10] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, "TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks," in *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN '06)*, pp. 63–70, May 2006.

[11] E. Ertin, A. Arora, R. Ramnath et al., "Kansei: a testbed for sensing at scale," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 399–406, April 2006.

[12] P. Gireesan Namboothiri, A. Anand Kumar, K. M. Sivalingam, and S. A. V. Satya Murty, "A testbed for distributed target tracking with directional sensors," in *Proceedings of the IFIP Wireless Days Conference*, pp. 10–12, Niagara Falls, Canada, October 2011.

[13] J. Schiller, A. Liers, and H. Ritter, "ScatterWeb: a wireless sensornet platform for research and teaching," *Computer Communications*, vol. 28, no. 13, pp. 1545–1551, 2005.

[14] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: a wireless sensor network testbed," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 483–488, April 2005.

[15] A. Kuzu, S. Erboral, S. Bogosyan, and M. Gokasan, "Multipurpose wireless sensor network platform for research and training in data fusion, and multi-feature target tracking," in *Proceedings of the 2nd Conference on Human System Interactions (HSI '09)*, pp. 184–189, May 2009.

[16] J. P. Sheu, C. J. Chang, C. Y. Sun, and W. K. Hu, "WSNTB: a testbed for heterogeneous wireless sensor networks," in *Proceedings of the the 1st IEEE International Conference on Ubi-Media Computing and Workshops (U-Media '08)*, pp. 338–343, August 2008.

[17] L. Wu, Y. Xu, C. Xu, and F. Wang, "Plug-configure-play service-oriented gateway for fast and easy sensor network application development," in *Proceedings of the 2nd International Conference on Sensor Networks (SENSORNETS '13)*, 2013.

[18] http://www.ti.com.cn/product/cc2530.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 192128, 10 pages
http://dx.doi.org/10.1155/2013/192128

Research Article

Time Synchronization in Wireless Sensor Networks Using Max and Average Consensus Protocol

Zhao Dengchang, An Zhulin, and Xu Yongjun

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Correspondence should be addressed to Xu Yongjun; xyj@ict.ac.cn

Received 26 December 2012; Accepted 28 February 2013

Academic Editor: Guangjie Han

This paper proposes a novel distributed time synchronization scheme for wireless sensor networks, which uses max consensus to compensate for clock drift and average consensus to compensate for clock offset. The main idea is to achieve a global synchronization just using local information. The proposed protocol has the advantage of being totally distributed, asynchronous, and robust to packet drop and sensor node failure. Finally, the protocol has been implemented in MATLAB. Through several simulations, we can see that this protocol can reduce clock error to ±10 ticks, adapt to dynamic topology, and be suitable to large-scale applications.

1. Introduction

As in all distributed systems, time synchronization is very important in wireless sensor networks (WSNs) since the design of many protocols and implementation of applications require precise time, for example, forming an energy-efficient radio schedule, conducting in-network processing (data fusion, data suppression, data reduction, etc.), distributing an acoustic beamforming array, performing acoustic ranging (i.e., measuring the time of flight of sound), logging causal events during system debugging, and querying a distributed database.

Time synchronization is a research area with a very long history. Various mechanisms and algorithms have been proposed and extensively used over the past few decades. However, several unique characteristics of WSNs often preclude the use of the existing synchronization techniques in this domain. First, since the amount of energy available to battery-powered sensors is quite limited, time synchronization must be implemented in an energy-efficient way. Second, some messages need to be exchanged for achieving synchronization while limited bandwidth of wireless communication discourages frequent message exchanges among sensor nodes. Third, the small size of a sensor node imposes restrictions on computational power and storage space. Therefore, traditional synchronization schemes such as network time protocol (NTP) and global positioning system (GPS) are not suitable for WSNs because of complexity and energy issues, cost efficiency, limited size, and so on.

In the context of WSNs, time synchronization refers to the problem of synchronizing clocks across a set of sensor nodes that are connected to one another over a single-hop or multihop wireless networks. To achieve time synchronization in wireless sensor networks, we have to face the following four challenges.

1.1. Nondeterministic Delays. There are many sources of message delivery delays. Kopetz and Ochsenreiter [1] describe the components of message latency, which they call the Reading Error, as being comprised of 4 distinct components plus the local granularity of the nodes clocks. Their work was later expanded by [2] to include transmission and reception time. The most nondeterministic delay is called Access Time, which is incurred in the MAC layer waiting for access to the transmit channel, its orders of magnitude is larger than the synchronization precision required by the network.

1.2. Clock Drift. Manufacturers of crystal oscillators specify a tolerance value in parts per million (PPM) relative

the nominal frequency at 25°C, which determines the maximum amount that the skew rate will deviate from 1. For the nodes used in WSNs, the tolerance value is typically in the order of 5 to 20 PPM. If no drift compensation applied, two synchronized nodes will be out of step soon.

1.3. Robustness. Since sensor networks are often left unattended for long periods of time in possibly hostile environments, synchronization schemes should be robust against link and node failures. Mobile nodes can also disrupt routing schemes, and network partitioning may occur.

1.4. Convergence Speed. Nodes in wireless sensor networks always distribute in large scales, one node may get in touch with another by many hops. This increases the difficulty in reducing the convergence speed in time synchronization algorithm design.

Up to now, many protocols have been designed to address this problem. These protocols all have some basic features in common: a simple connectionless messaging protocol, exchange of clock information among nodes, mitigating the effect of nondeterministic factors in message delivery, and processing utilizing different schemes and algorithms, respectively. They can be classified into two types: centralized synchronization and distributed synchronization.

Centralized synchronization protocol, such as RBS [4], TPSN [2], and FTSP [3], usually has fast convergence speed and little synchronization error. This kind of protocol needs a physical node acting as the whole network's reference clock, so it has to divide the nodes into different roles, for example, client node and beacon node in RBS. If the node with the special role, such as beacon node in RBS, is out of work, the protocol will suffer from big damage. To deal with the WSNs' dynamic topology, centralized synchronization protocol is often designed with complexity logic. Another disadvantage of centralized synchronization protocol is that synchronization error grows with the increase of network hops.

Distributed synchronization protocol, such as TDP [5]/GCS [6]/RFA [7]/ATS [8]/CCS [9], can use local information to achieve the whole network synchronization. This kind of protocol can easily adapt to WSNs' dynamic topology property with lite computation. Currently, the disadvantage of distributed synchronization protocol is that the convergence speed may be a bit slow, relating to the network topology.

This paper describes a new distributed protocol for time synchronization in wireless sensor networks called time synchronization using max and average consensus protocol (TSMA). We adapt a number of techniques to take up the challenges time synchronization has in WSNs. To eliminate the nondeterministic delays, we make use of MAC layer timestamp technique. To compensate for the clock drift, we adapt max consensus protocol, and we use average consensus protocol to compensate for the clock offset. This protocol has the advantages of being computationally light, scalable, asynchronous, robust to node and link failure, and it does not require a master or controlling node.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 introduces some mathematical tools and definitions that will be instrumental for the proof of convergence of the proposed TSMA algorithm. Section 4 introduces a model for the clock dynamics and formally defines the synchronization objectives, while Section 5 presents the TSMA algorithm in details. This is followed by MATLAB simulations in Section 6. Finally, Section 7 briefly summarizes the results obtained and proposes potential research directions.

2. Related Work

Typical time synchronization algorithms in WSNs include timing-sync protocol for sensor networks (TPSNs) [2], reference broadcast synchronization (RBS) [4], flooding time synchronization protocol (FTSP) [3], time diffusion protocol (TDP) [5], global clock synchronization (GCS) [6], Reach-back Firefly algorithm (RFA) [7], average time synchronization (ATS) [8], and consensus clock synchronization (CCS) [9]. RBS, TPSN, and FTSP are centralized synchronization protocols, while TDP, GCS, RFA, ATS, and CCS are distributed synchronization protocols.

TPSN [2] is designed as a flexible extension of NTP [10] for use in wireless sensor networks, which consists of two phases: a level discovery phase and a synchronization phase. The level discovery phase organizes the whole network into a hierarchical tree topology with the master node at its root, then perform the pair wise synchronization along the branches of the hierarchical structure using the classical sender-receiver synchronization handshake exchange in the synchronization phase. TPSN is a centralized synchronization protocol, utilizes MAC layer timestamping to reduce message delivery nondeterminism and improve synchronization accuracy, its convergence speed increases linearly with the max network hops. This approach suffers from two limitations. The first limitation arises because if the root node or parent node dies, then a new root election or parent-discovery procedure needs to be initiated, thus adding substantial overhead to the code and long periods of network desynchronization. The second limitation is due to the fact that geographically closed nodes might be far in terms of the tree distance, which is directly related to increase the clocks error locally.

RBS [4] aims to provide synchronization amongst a set of client nodes located within the single-hop broadcast range of a beacon node. Compared to the traditional protocols working on an LAN, its main contribution is to directly remove two of the largest sources of nondeterminism involved in message transmission, transmission time, and access time, by exploiting the concept of a time critical path, which is the path of a message that contributes to nondeterministic synchronization errors. This centralized protocol uses least squares linear regression to compensate for the clock drift, its convergence speed also increases linearly with the max network hops. Using the beacon node makes this protocol vulnerable to node failure and node mobility.

FTSP [3] is an ad-hoc, multihop time synchronization protocol for WSNs. It achieves high accuracy by utilizing timestamping of radio messages in low layers of radio stack, completely eliminating access time to the radio channel required in CSMA MAC protocols. Further accuracy is achieved by compensating for clock skews of participating nodes via linear regression. Several mechanisms are used to provide robustness against node and link failures, most notably periodic flooding of synchronization messages throughout the whole network, but this approach still do not completely solve the limitations aforementioned in TPSN.

Nodes employing TDP [5] periodically self-determine to become master/diffusion leader nodes using an election/re-election procedure. Master nodes then engage neighboring nodes in a peer evaluation procedure to isolate problem nodes. Timing information messages are broadcasted from master nodes and then rebroadcasted by the diffused leader nodes for a fixed number of hops, forming a radial tree structure. This approach is fully distributed, but it does not compensate for the clock drift. To the application needing small tolerance time, this algorithm has to increase the synchronization frequency, which greatly decreases the node's battery life.

In GCS [6], nodes take turns to broadcast a synchronization request to their neighbors who each respond with a message containing their local time. The receiving node (at the center of this exchange) averages the received timestamps and broadcasts this value back to its neighbors which adopt this value as their new time. This is repeated by each node in the network until network wide synchronization is achieved. This approach is fully distributed, but it does not compensate for the clock drift. To get higher time accuracy, it has to decrease the time synchronization period.

RFA [7] is inspired by firefly synchronization mechanism. In this algorithm, every node periodically broadcasts a synchronization message, and anytime they hear a message they advance, of a small quantity, the phase of their internal clock that schedules the periodic message broadcasting. Eventually all nodes will advance their phase till they are all synchronized, that is, they fire a message at the same time. This approach, however, does not compensate for clock drift, therefore the firing period needs to be rather small.

ATS [8] uses a cascade of two consensus algorithms to tune compensation parameters and converge nodes to a steady state virtual clock. In the first stage, nodes broadcast local timestamps in order to estimate the clock skew rates relative to each other. Nodes then broadcast their current estimate of the virtual clock skew rate and receiving nodes combine this with their relative skew estimates to adjust their own virtual clock estimate. The same principle is then applied to remove the offset errors. This is a fully distributed protocol; its convergence speed is a bit slow, related to the network topology.

CCS [9] utilizes average consensus algorithm to compensate the clock offset. By the accumulated offset error that they remove during each round of offset compensation nodes can observe how much their own clocks drift away from the consensus time, then they use this information to compensate the clock drift, which can be seen as an enhancement of ATS. This approach is also fully distributed and has the disadvantage of slow convergence speed.

Table 1: Time synchronization features for different protocols.

	Distrib.	Skew comp.	MAC timestamp	Linear skew convergence speed
TPSN [2]	No	No	No	No
FTSP [3]	No	Yes	Yes	Yes
RBS [4]	No	Yes	Yes	Yes
TDP [5]	Yes	No	No	No
GCS [6]	Yes	No	No	No
RFA [7]	Yes	No	No	No
ATS [8]	Yes	Yes	Yes	No
CCS [9]	Yes	Yes	Yes	No
TSMA	Yes	Yes	Yes	Yes

The TSMA protocol proposed in this paper is similar to ATS and CCS. It is fully distributed, that is, it does not require any special root, including skew compensation, and exploits MAC-layer timestamping for higher accuracy. Unlike ATS and CCS, the proposed protocol uses max consensus method to compensate the clock drift, which has faster skew convergence speed as the centralized protocol, that is, it increases linearly with the max network hops. The features of all the protocols mentioned in this section are summarized in Table 1.

3. Mathematical Preliminaries

We assume that links in WSNs are symmetric, so the network can be modeled as an undirected graph $G = (V, E)$, where $V = \{v_i \mid i = 1, 2, \ldots, n\}$ represents the nodes in the WSNs, and the edge set E represents the available communication links, that is, $(v_i, v_j) \in E$ if node j sends information to node i. We refer to v_i and v_j as the tail and head of the edge (v_i, v_j), respectively. The orientation of the graph is a choice of heads and tails for each undirected edge. The set of edges of a fix orientation of the graph is denoted by E_0. Thus, E_0 contains one and only one of the two edges (v_i, v_j), $(v_j, v_i) \in E$. We use m to present edge number and n the vertex number, for example, $n = |V|$, $m = |E_0|$. The set of neighbors of node is denoted by $N_i = \{v_j \mid e_{i,j} \in E\}$. The degree of node v_i is the number of its neighbors $|N_i|$ and is denoted by $\deg(v_i)$. The degree matrix is an $n \times n$ matrix defined as $\Delta = \Delta(G) = \{\Delta_{e_{i,j}}\}$, where

$$\Delta_{e_{i,j}} = \begin{cases} \deg(v_i), & i = j \\ 0, & i \neq j. \end{cases} \quad (1)$$

Let A denote the adjacency matrix of G. Then, the Laplacian of graph G is defined by

$$L = \Delta - A. \quad (2)$$

The matrix L has the following features.

(i) All the row sums of L are zero, and thus $e_0 = (1, 1, \ldots, 1)^T \in R^n$ is an eigenvector of L associated with the eigenvalue 0.

(ii) Fix and orientation of the graph, and let $E_0 = \{e_i \mid i = 1, 2, \ldots, m\}$, then $L = CC^T$, where $C = \{c_{pq}\}$ is an $n \times m$ matrix, n is the number of vertex set V, and m is the number of the edge set E_0,

$$c_{pq} = \begin{cases} +1, & \text{if } v_p \text{ is the head of edge } e_q; \\ -1, & \text{if } v_p \text{ is the tail of edge } e_q; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

It is a well-known fact that this property holds regardless of the choice of the orientation of G [11]. Let x_i denote a scalar real value assigned to v_i. Then, $x = (x_1, x_2, \ldots, x_n)^T$ denotes the state of the graph G. We define the Laplacian potential of the graph as follows:

$$\Psi_G(x) = \frac{1}{2} x^T L x. \tag{4}$$

Lemma 1 (Laplacian potential [12]). *$\Psi_G(x)$ has the following properties.*

(i) *Undirected graph G's Laplacian potential is positive semidefiniteness, and*

$$x^T L x = \sum_{(v_i, v_j) \in E_0} \left(x_i - x_j\right)^2. \tag{5}$$

(ii) *If G is a connected graph, $\Psi_G(x) = 0$ if and only if $x_i = x_j$, for all i, j.*

Lemma 2 (connectivity and graph Laplacian [11]). *Assume graph G has c connected components, then*

$$\operatorname{rank}(L) = n - c. \tag{6}$$

Particularly, for a connected graph with $c = 1$, $\operatorname{rank}(L) = n - 1$.

Lemma 3. *G is an undirected and connected graph, L is G's Laplacian matrix, then there must be a zero in L's eigenvalues, and all the other eigenvalues of G are positive and real.*

Proof. G is an undirected graph, then L is a real symmetric matrix. After diagonalization, the rank of matrix L stays the same. G is a connected graph, then based on Lemma 2, the rank of matrix L is $n - 1$, then there must be a zero in L's eigenvalues, and all the other eigenvalues of G are nonzero. L is a real symmetric matrix, so L's eigenvalues are real. Based on Lemma 1, L is a positive semidefinite matrix, then L's eigenvalues are nonnegative. In conclusion, there must be a zero in L's eigenvalues, and all the other eigenvalues of G are positive and real. □

Definition 4 (consensus). Let the value of all nodes x be the solution of the following differential equation:

$$\dot{x} = f(x), \quad x(0) \in R^n. \tag{7}$$

In addition, let $X : R^n \to R$ be a multi-input single-output operation on $x = (x_1, x_2, \ldots, x_n)^T$ that generates a decision value $y = X(x)$. We say all the nodes of the graph have reached consensus w.r.t. X in finite time $T > 0$ if and only if all the nodes agree and $x_i(T) = X(x(0))$, for all $i \in I$. Some of the common examples of the operation X are given as follows:

(i) average consensus,

$$X(x) = \operatorname{Ave}(x) = \frac{1}{n} \sum_{i=1}^{n} x_i; \tag{8}$$

(ii) max consensus,

$$X(x) = \operatorname{Max}(x) = \max\left(x_1, x_2, \ldots, x_n\right); \tag{9}$$

(iii) min consensus,

$$X(x) = \operatorname{Min}(x) = \min\left(x_1, x_2, \ldots, x_n\right). \tag{10}$$

Theorem 5. *G is an undirected and connected graph if every node in G runs the following distributed protocol:*

$$u_i(t) = \sum_{j \in N_i} \left(x_j(t) - x_i(t)\right) + T_0, \tag{11}$$

where T_0 is a constant, then the vector of the value of the nodes x is the solution of the following ordinary differential equations:

$$\dot{x} = -Lx + \left(T_0, T_0, \ldots, T_0\right)^T, \quad x(0) \in R^n. \tag{12}$$

In addition, all the nodes of the graph globally asymptotically reach an average consensus, and then every node will have the same status,

$$x_i(t) = x_j(t) = tT_0 + \operatorname{Ave}(x(0)), \quad \forall i, j, \ i \neq j. \tag{13}$$

Proof. Equation (12) corresponds to a typical linear time invariant system; its complete response can be solved by calculating zero-status response and zero-input response, respectively.

(a) Solve the zero-status response $x_s(t)$, where

$$\dot{x}_s = -Lx_s + \left(T_0, T_0, \ldots, T_0\right)^T, \qquad x_s(0) = (0, 0, \ldots, 0)^T. \tag{14}$$

Based on Laplacian matrix L's features, the particular solution of $x_s(t)$ is $x_p(t) = T_0(t, t, \ldots, t)^T$. $-L$ is a real symmetric matrix, so $-L$ can do diagonalization, let P be the corresponding invertible matrix. Based on the theory of linear algebra, the homogeneous solution of $x_s(t)$ is

$$x_h(t) = c_1\alpha_1 e^{\lambda_1 t} + c_2\alpha_2 e^{\lambda_2 t} + \cdots + c_n\alpha_n e^{\lambda_n t}, \tag{15}$$

where c_i $(1 \le i \le n)$ is undetermined coefficient, λ_i $(1 \le i \le n)$ is matrix L's ith eigenvalue, α_i $(1 \le i \le n)$ is P's ith column, which is the corresponding eigenvector of eigenvalue λ_i, so the zero-status's complete response is

$$\begin{aligned} x_s(t) = x_p(t) + x_h(t) &= T_0(t, t, \ldots, t)^T \\ &+ c_1\alpha_1 e^{\lambda_1 t} + c_2\alpha_2 e^{\lambda_2 t} + \cdots + c_n\alpha_n e^{\lambda_n t}. \end{aligned} \tag{16}$$

We know that $x(0) = (0, 0, \ldots, 0)^T$, so $c_1\alpha_1 + c_2\alpha_2 + \cdots + c_n\alpha_n = (0, 0, \ldots, 0)^T$, in addition, $\alpha_1, \alpha_2, \ldots, \alpha_n$ is linear independence, then $c_1 = c_2 = \cdots = c_n = 0$, the solution of zero-status response is $x_s(t) = T_0(t, t, \ldots, t)^T$.

(b) Solve the zero-input response $x_s(t)$, where

$$\dot{x}_0 = -Lx_0, \quad x_0(0) = x(0). \tag{17}$$

Observe that all the eigenvalues of $-L$ are negative, except for the single zero eigenvalue. Thus, any solution of the system asymptotically converges to a point x_0^* in the eigenspace associated with $\lambda_1 = 0$. This implies that an average consensus is globally asymptotically reached by all the nodes. Let x_0^* be one equilibrium of the system $\dot{x}_0 = -Lx_0$. Then, $-Lx_0^* = 0$, and thus x^* is the eigenvector of the Laplacian L associated with the zero eigenvalue $\lambda_1 = 0$. On the other hand, we have $\psi_G(x_0^*) = (1/2)(x_0^*)^T L x_0^* = 0$, and from connectivity of G, it follows that $x_{0,i}^* = x_{0,j}^* = a$, for all i, j, that is, $x_0^* = (a, a, \ldots, a)^T$, $a \in R$. Notice that $\sum_{i=1}^n x_{0,i} = 0$. Thus, $\mathrm{Ave}(x_0)$ is an invariant quantity. This implies $\mathrm{Ave}(x_0^*) = \mathrm{Ave}(x(0))$. But $\mathrm{Ave}(x_0^*) = a$, therefore $a = \mathrm{Ave}(x(0))$, then the solution of zero-input response is $x_0^* = (a, a, \ldots, a)^T$, where $a = \mathrm{Ave}(x(0))$.

Above all, when zero-input response reaches the equilibrium status, the system's full response is

$$\begin{aligned} x(t) &= x_s(t) + x_0(t) \\ &= \left(tT_0 + \mathrm{Ave}(x(0)),\ tT_0 + \mathrm{Ave}(x(0)), \ldots, tT_0 \right. \\ &\quad \left. + \mathrm{Ave}(x(0))\right)^T. \end{aligned} \tag{18}$$

In other words, every node in the network finally has the same status. □

4. Clock Model

Clock synchronization is of significant importance in WSNs. Before delving into the details of synchronizing clocks, we first define some terminologies used in this paper. $c(t)$ denotes a reference clock, where t is the real time, that is, coordinated universal time (UTC). Every sensor node maintains its own local clock, which is a monotonically nondecreasing function of t. This local clock is an ensemble of hardware and software components, essentially a timer that counts the oscillations of a quartz crystal running a particular frequency. In general, the timer is programmed to generate an interrupt, which is called a clock tick. At each clock tick, the interrupt procedure increments the clock value stored in memory.

For any two nodes' local clocks $c_i(t)$ and $c_k(t)$, the clock $c_i(t)$ is considered correct at time t if $c_i(t) = c(t)$, or the clock $c_i(t)$ is considered accurate at time t if $dc_i(t)/dt = dc(t)/dt$, and two clocks $c_i(t)$ and $c_k(t)$ are synchronized at time t if $c_i(t) = c_k(t)$. The aforementioned definitions show that the two synchronized clocks are not always correct or accurate; time synchronization is not related to time correctness and accuracy. As for most applications in WSNs, it is sufficient to achieve clock synchronization. Since the oscillator frequency is time varying due to ambient conditions such as temperature changes, variations of electric supply voltage, and air pressure, clock $c_i(t)$ of node i can be modeled as

$$c_i(t) = k_i t + b_i, \tag{19}$$

where b_i is the clock offset, the difference between the time reported by clock $c_i(t)$ and the real time at the initial instant t_0 (i.e., $b_i = c_i(t_0) - c(t_0)$), and k_i is defined as the skew of the clock $c_i(t)$. To achieve perfect synchronization in a sensor network, all nodes $i = 1, 2, \ldots, N$ must precisely compensate for their clock parameters k_i and b_i so that all clocks have zero offset error and all tick at the same rate.

5. TSMA Method

In this section, we present our time synchronization algorithm, that is, TSMA. Instead of trying to synchronize to an external reference like absolute time or UTC, the TSMA method aims to achieve an internal consensus within the network on what time is, and how fast it travels. In every synchronization round, the TSMA algorithm updates the compensation parameters for each node and over time the network clocks converge to a consensus,

$$\lim_{t \to \infty} c_i'(t) = c_c(t), \tag{20}$$

where $c_c(t)$ is the consensus clock.

The consensus clock is not a physical clock, it is a new virtual clock, that is, generated from the network of nodes running the TSMA algorithm. The consensus clock has its own skew rate and offset relative to the absolute time.

By expanding the clock functions from both sides of (20), we can see the compensation parameters that all nodes must obtain in order to synchronize to the consensus clock,

$$\begin{aligned} \lim_{t \to \infty} \left(k_i' c_i(t) + b_i'\right) &= k_c t + b_c, \\ \lim_{t \to \infty} \left(k_i' k_i t + k_i' b_i + b_i'\right) &= k_c t + b_c. \end{aligned} \tag{21}$$

Then, we have

$$\begin{aligned} \lim_{t \to \infty} k_i' &= \frac{k_c}{k_i}, \\ \lim_{t \to \infty} b_i' &= b_c - k_i' b_i. \end{aligned} \tag{22}$$

To achieve these compensation parameters, nodes repeat the TSMA algorithm in synchronization rounds which consist of two main tasks; skew compensation and offset compensation. In skew compensation, the nodes iteratively select the max skew rate estimation from its neighbors in order to improve their skew compensation parameter. In the offset compensation phase, nodes average their neighbors' clock readings to synchronize nodes to a common time.

5.1. Skew Compensation. The goal of skew compensation is to ensure all compensated clocks in the network tick at the same rate, that is,

$$\lim_{t \to \infty} k_i' k_i = k_c, \quad \forall i. \tag{23}$$

We use max consensus protocol to compensate for clock skew. For perfect skew compensation, the skew rate of the consensus clock is given by

$$k_c = \max\left(k_1, k_2, \ldots, k_n\right). \tag{24}$$

Nodes execute the following algorithm to converge their clock skew to the consensus value.

Step 1. Before running the synchronization algorithm, nodes set their skew estimation k_i' to 1, and set their respect life time life_i to 0.

Step 2. At the beginning of each synchronization round, every node increases their own life time life_i by 1. If node i's life time $\text{life}_i > 3$, then node i broadcasts a synchronization packet containing its own local clock value τ_i and estimated skew k_i' using a CSMA/CA MAC scheme once per synchronization round. Like other synchronization techniques, MAC layer timestamping is required to ensure that clock values are transmitted instantly.

Step 3. If node i receives a synchronization packet from node j at time t, then this node should immediately record its own local time $\tau_i(t)$ and node j's information $(\tau_j(t), k_j'(t))$. If node i has not stored a history local time pair before, then create the local time pair $(\tau_i(\text{old}), \tau_j(\text{old}))$ and assign it with

$$\left(\tau_i\left(\text{old}\right), \tau_j\left(\text{old}\right)\right) = \left(\tau_i\left(t\right), \tau_j\left(t\right)\right). \tag{25}$$

If the history local time pair has already existed, and satisfies $k_i'(t) * (\tau_j(t) - \tau_j(\text{old})) > k_j'(t) * (\tau_i(t) - \tau_i(\text{old}))$, we will update this node's estimated skew k_i' and the history local time pair $(\tau_i(\text{old}), \tau_j(\text{old}))$ using

$$\begin{aligned} k_i'\left(t\right) &= k_j'\left(t\right) * \frac{\tau_i\left(t\right) - \tau_i\left(\text{old}\right)}{\tau_j\left(t\right) - \tau_j\left(\text{old}\right)}, \\ \left(\tau_i\left(\text{old}\right), \tau_j\left(\text{old}\right)\right) &= \left(\tau_i\left(t\right), \tau_j\left(t\right)\right). \end{aligned} \tag{26}$$

5.2. Offset Compensation. After the skew compensation algorithm is applied, all local estimators will eventually have the same drift, that is, they will run at the same speed. At this point, it is only necessary to compensate for possible offset errors. To achieve this, each node aims to accurately estimate the instantaneous average time of all its neighbors' clock in the network and set their clocks to this time. From Theorem 5, we can see that every node in the network will finally reach the same status,

$$\bar{b}\left(t\right) = \frac{1}{\left|N_i\right|} \sum_{j \in N_i} b_j\left(t\right). \tag{27}$$

In reality, however, it is not practically possible for nodes to calculate the instantaneous global average time of all its neighbors' clock in the network. Since sensor networks may consist of a large number of nodes, and one node may have lots of neighbors. It is impossible for nodes to exchange clock values instantaneously, and nodes may suffer from packet loss and other network dynamics.

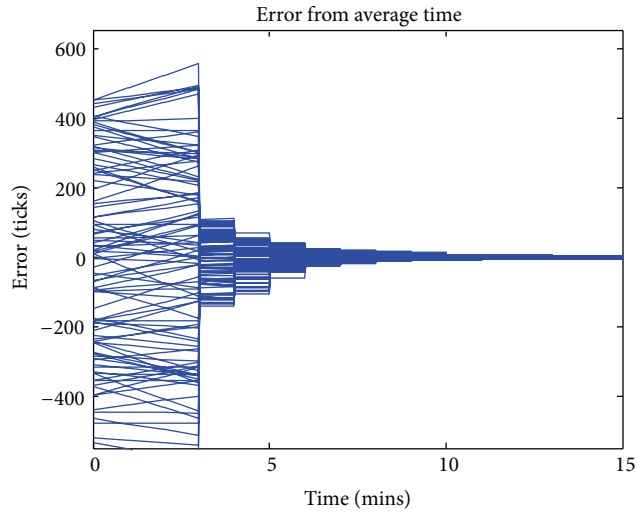


Figure 1: Clock error of 100 simulated nodes running the TSMA protocol in a $10 * 10$ network with synchronization period of 1 min and transmission range of 2.

Instead, the TSMA algorithm converges network clocks to an approximation of $\bar{b}(t)$ using the locally available information within the single-hop transmission range of each node. Nodes execute the following algorithm to converge their clock offset to the consensus value.

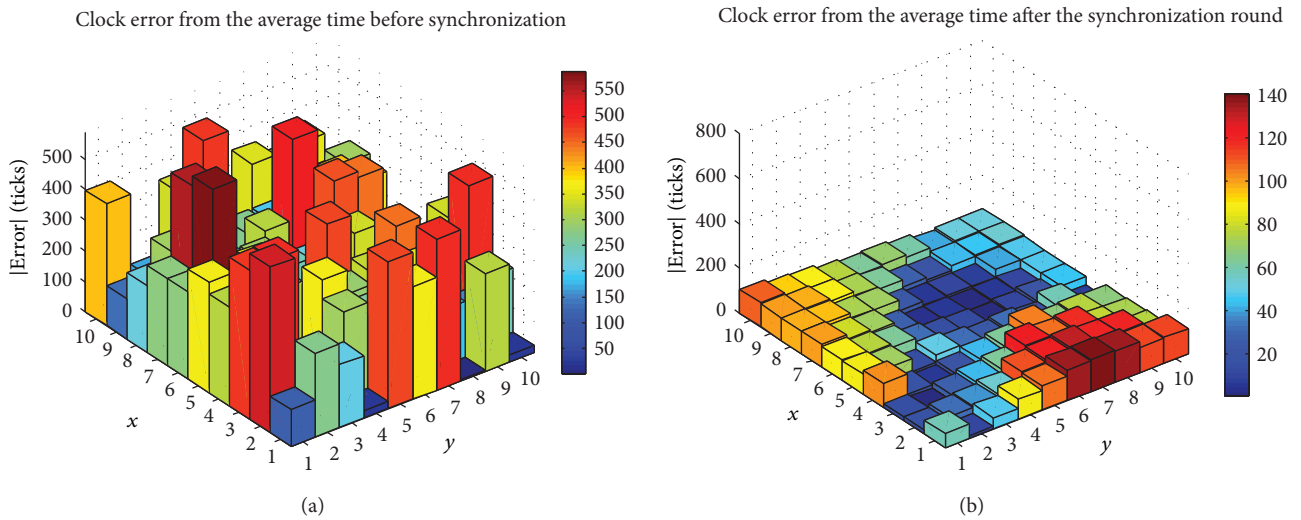
Step 1. Before running the synchronization algorithm, nodes set their offset estimation b_i' to their current local time and set their respect life time life_i to 0.

Step 2. At the beginning of each synchronization round, nodes reset their confidence parameter Conf_i to 1 and increase their life time life_i by 1. If node i's life time $\text{life}_i > 3$, then node i broadcasts a synchronization packet containing its estimated offset b_i' and confidence parameter Conf_i using a CSMA/CA MAC scheme once per synchronization round. MAC layer timestamping is required to ensure that clock values are transmitted instantly.

Step 3. For each broadcast, all other nodes within receiving range update their own clocks by combining this value with their own clocks using the confidence weighted running average equation and then increase their confidence parameter Conf_i by 1,

$$\begin{aligned} b_i'\left(t\right) &= \begin{cases} b_j'\left(t\right), & \text{if } \text{life}_i = 1 \\ \dfrac{\text{Conf}_i(t) * b_i'(t) + \text{Conf}_j(t) * b_j'(t)}{\text{Conf}_i(t) + \text{Conf}_j(t)}, & \text{if } \text{life}_i > 1, \end{cases} \\ \text{Conf}_i\left(t\right) &= \text{Conf}_i\left(t\right) + 1. \end{aligned} \tag{28}$$

The confidence parameters give weighting to the clock values from nodes that have already received synchronization packets in the current round. These nodes are generally closer to a local consensus than those who have not received



(a) (b)

Figure 2: Geographic distribution of synchronization error of a 10 ∗ 10 sensor network for transmission range of 2 (a) before and (b) after one round of offset compensation.

any sync packets, and this was found to increase the rate of convergence.

6. Simulation

In this paper, we use max consensus to compensate for clock skew and use average consensus to compensate for clock offset. To show the correctness and robustness of the TSMA algorithm, we make a simulation in MATLAB; the following is the simulation environment: there are 100 WSN nodes in the 10 ∗ 10 grid, one grid one node, and we place the node in the center of the grid, the base value of nodes' oscillator is 32.768 KHz, the skew rates are assigned randomly from a normal distribution with mean = 1 and standard deviation = 20 PPM, nodes are given an initial error distribution between 0 and 1000 ticks.

6.1. Offset Compensation. In this simulation, we set the time synchronization period to 1 minute. Nodes broadcast one sync packet per round in random order with a transmission range of 2, which means if the distance between two nodes is less than 2, then they can communicate with each other. Figure 1 shows the clock errors from average time measured in clock cycles. In the first three synchronization periods, nodes' life time is less than 3, so they do not transmit any synchronization packet to their neighbors, and then in 3 minutes, nodes start to compensate for the clock skew and offset. After seven rounds compensation, 100 nodes in the network achieve the synchronization state, the synchronization error between any two nodes is included between ±10 ticks.

The geographic distribution of clock errors in the network is shown in Figure 2. It can be seen that before Figure 2(a), the first round of offset compensation, the clock error relative to the average time can reach up to 550 ticks, but after Figure 2(b), the first round offset compensation, the clock error reduces to 140 ticks. Also we can see that the clock error after compensation transfer smoothly in geographic distribution, the local error between any neighboring nodes is no larger than 50 ticks.

A histogram of the clock errors is shown in Figure 3. It can be seen that before Figure 3(a), the first round of offset compensation, the standard deviation of clock error reaches up to 286.32, but after Figure 3(b), the first round of offset compensation, this value reduces to 70.90. From Figures 2 and 3, we can see that after the first round of offset compensation, the clock errors among nodes are significantly reduced.

Figure 4 shows the convergence of the skew rates of 100 nodes in this simulation. From Figure 4, we can see that after two synchronization periods all the 100 nodes' skew rates converge on the maximum clock skew.

6.2. Transmission Range. Figure 5 shows the transmission range's effect to algorithm performance. We set the transmission range to 1, 2, and 3, respectively, then compare their convergence speed with each other. From Figure 5, we can see that synchronization accuracy and convergence speed are found to improve as the transmission range of the nodes increased, since nodes could receive synchronization packets from a greater proportion of the network population. On the other side, to increase the transmission range, we also increase the energy consumption of nodes, so in the practical application, we should make a tradeoff between energy consumption and convergence speed.

6.3. Dynamic Topology. The simulation, shown in Figure 6, is intended to study the robustness properties of the TMSA protocol subject to node failure and node replacement, as well as the performance in terms of convergence speed and steady state synchronization error. The synchronization period is set to 1 minute which is sufficiently large to exhibit the effects of different clock speeds. The simulation was run for about

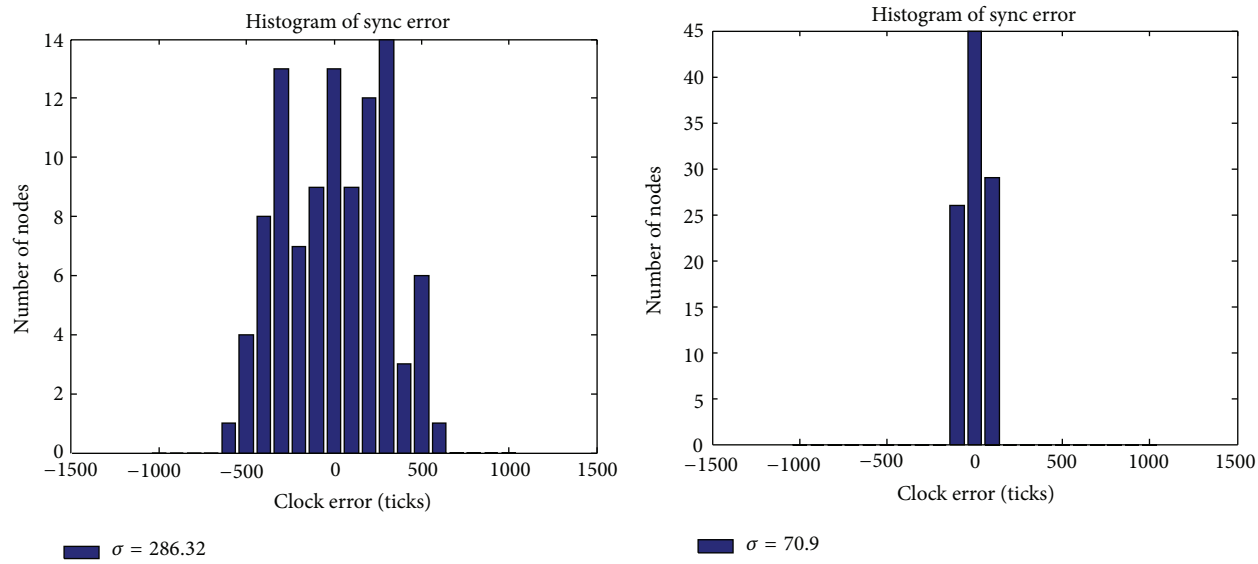


Figure 3: Histogram of the synchronization errors from the instantaneous global average time of the sensor network (a) before and (b) after one round of offset compensation.

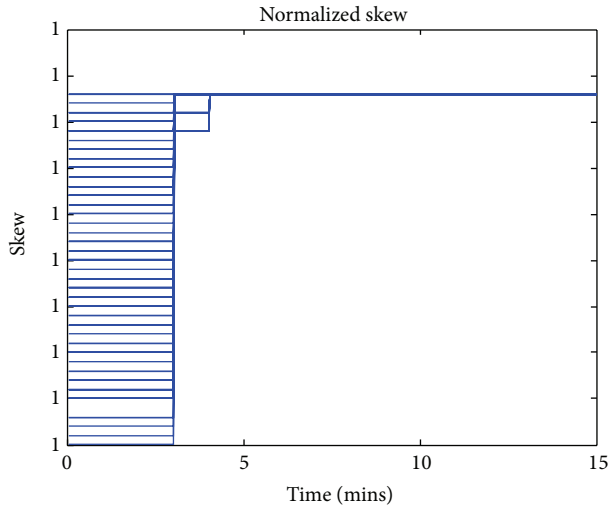


Figure 4: Convergence of the skew rate of 100 simulated nodes running the TSMA protocol in a 10 ∗ 10 network with synchronization period of 1 min and transmission range of 2.

50 minutes and presents 4 different regions of operation indicated by the letters A, B, C, and D which model potential node failure or the addition of the new nodes. In Region A, all nodes are turned on simultaneously with random initial conditions of their local clocks. After 2 synchronization periods, the synchronization error between any two nodes is included between ±10 ticks, that is, the maximum error is smaller than 20 ticks. At the beginning of Region B, about 20% of the nodes chosen random in the grid are switched off and then switched on at different random times. Once a node is switched on, it starts updating its estimated time using TSMA protocol but does not transmit any message for the first three synchronization periods to avoid injecting large disturbances into the already synchronized network, and then it starts to transmitting and receiving messages equally. The plot in Figure 6 clearly shows that the nodes get synchronized as soon as they are turned on without perturbing the overall network behavior. At the beginning of Region C, about 20% of the nodes turned off their radio, that is, they stop updating their parameters k', b', then we turn on the 20% nodes' radio at about 35 minutes. In this procedure, we can see that all nodes keep the synchronized state, the reason is that their skew rates have already converged to the maximum skew rate in the network, these 20% nodes still synchronize with others. At the beginning of Region D, we replace the node in the top left corner with one node having 40 PPM drift, which is the max skew rate in the network now, and then all the other nodes will compensate themselves to this skew value. From Figure 6, we can see that three synchronization period after the new node joining the network, it starts to transmitting messages to its neighbors, after a short transient the nodes quickly synchronize again.

7. Conclusions

This paper presented a new synchronization algorithm for WSN, the time synchronization using consensus protocol, which is based on consensus algorithms whose main idea is to average local information to achieve a global agreement on a specific quantity of interest. The proposed algorithm is fully distributed, asynchronous, includes skew compensation, and is computationally light. Moreover, it is robust to dynamic network topologies due, for example, to node failure or replacement. Finally, a set of simulations are presented to show the good performance of the proposed protocol.

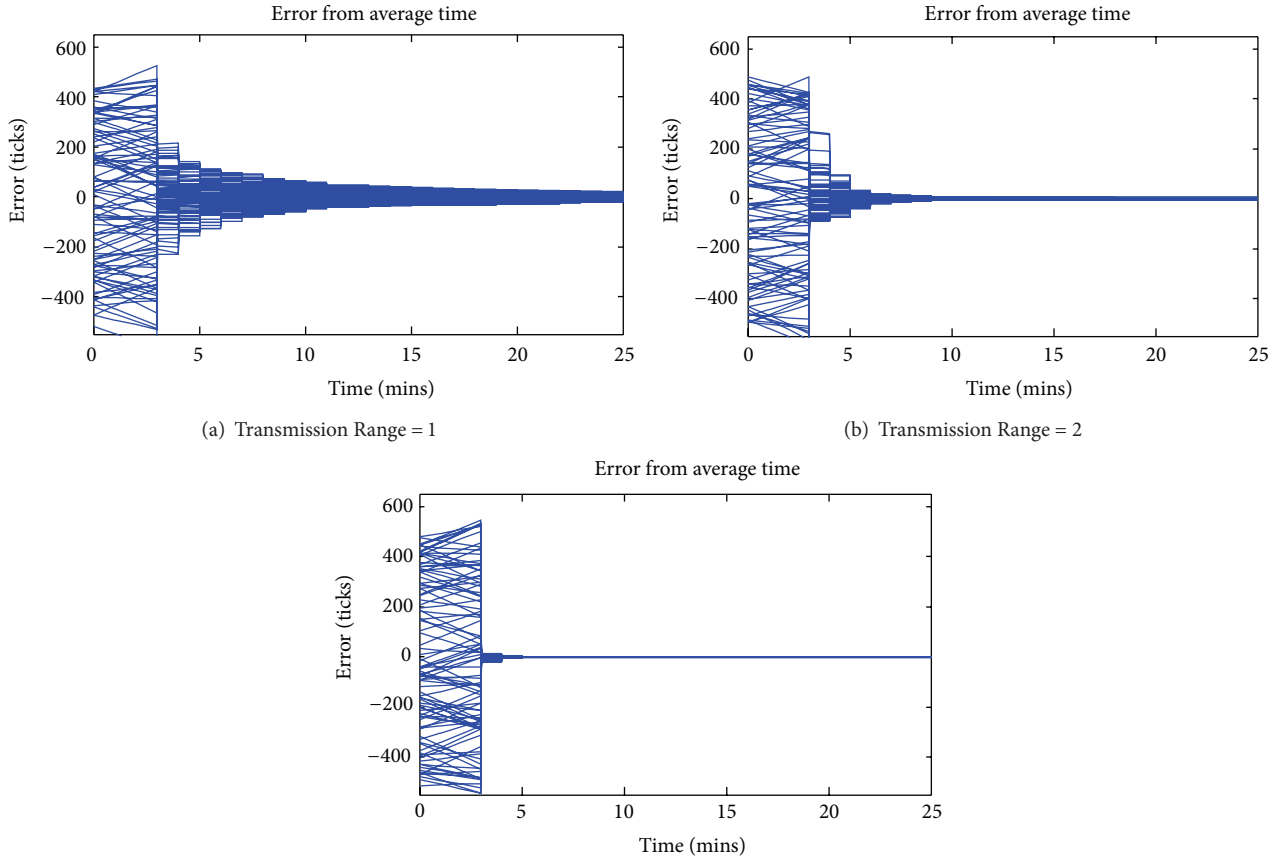


Figure 5: Clock errors of 100 simulated nodes running the TSMA protocol in a 10 ∗ 10 network with synchronization period of 1 min in transmission ranges of 1 (a), 2 (b), and 3 (c).

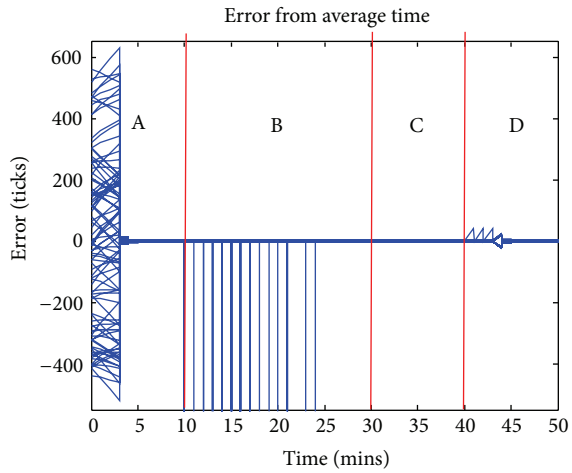


Figure 6: Clock errors of 100 simulated nodes running the TSMA protocol in a 10 ∗ 10 network with synchronization period of 1 min and transmission range of 3. Region A: all nodes are on. Region B: 20% of the nodes are turned off and then turned on at random times. Region C: 20% of the nodes turned off their radio, and then turned on their radio after 5 synchronization periods. Region D: replace the node in the top left corner with one node having 40 PPM drift.

However, extensive work is still necessary to compare the performance of our proposed approach relative to FTSP [3] and other protocols over large scale multihop sensor network and over longer periods.

Acknowledgments

This paper is supported in part by Important National Science & Technology Specific Projects under Grants nos. (2010ZX 03006-002, 2010ZX03006-007), National Basic Research Program of China (973 Program) (no. 2011CB302803), and National Natural Science Foundation of China (NSFC) under Grant no. (61173132,61003307). The authors alone are responsible for the content of the paper.

References

[1] H. Kopetz and W. Ochsenreiter, "Clock synchronization in distributed real-time systems," *IEEE Transactions on Computers*, vol. 36, no. 8, pp. 933–940, 1987.

[2] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st*

International Conference on Embedded Networked Sensor Systems (SenSys '03), pp. 138–149, Los Angeles, Calif, USA, November 2003.

[3] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *presented at Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, Md, USA, 2004.

[4] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Operating Systems Review*, vol. 36, pp. 147–163, 2002.

[5] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 384–397, 2005.

[6] L. Qun and D. Rus, "Global clock synchronization in sensor networks," in *Proceedings of the Conference on Computer Communications -23th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '04)*, pp. 564–574, March 2004.

[7] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *presented at the Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, San Diego, Calif, USA, 2005.

[8] L. Schenato and F. Fiorentin, "Average TimeSynch: a consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, pp. 1878–1886, 2011.

[9] M. K. Maggs, S. G. O'Keefe, and D. V. Thiel, "Consensus clock synchronization for wireless sensor networks," *IEEE Sensors Journal*, vol. 12, pp. 2269–2277, 2012.

[10] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[11] C. D. Godsil and G. Royle, *Algebraic Graph Theory*, Springer, 2001.

[12] R. Olfati Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the American Control Conference*, pp. 951–956, June 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 358267, 10 pages
http://dx.doi.org/10.1155/2013/358267

Research Article

Improvement and Performance Evaluation for Multimedia Files Transmission in Vehicle-Based DTNs

Di Wu,[1] Dongxia Zhang,[1] Guozhen Tan,[1] Limin Sun,[2] Juanjuan Li,[1] and Jiangchuan Liu[3]

[1] *School of Computer Science and Engineering, Dalian University of Technology, Dalian 116023, China*
[2] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*
[3] *School of Computer Science, Simon Fraser University, Vancouver, BC, Canada*

Correspondence should be addressed to Dongxia Zhang; lyzhangdongxia@126.com

Received 21 December 2012; Accepted 7 February 2013

Academic Editor: Hongsong Zhu

In recent years, P2P file sharing has been widely embraced and becomes the largest application of the Internet traffic. And the development of automobile industry has promoted a trend of deploying Peer-to-Peer (P2P) networks over vehicle ad hoc networks (VANETs) for mobile content distribution. Due to the high mobility of nodes, nodes' limited radio transmission range and sparse distribution, VANETs are divided and links are interrupted intermittently. At this moment, VANETs may become Vehicle-based Delay Tolerant Network (VDTNs). Therefore, this work proposes an Optimal Fragmentation-based Multimedia Transmission scheme (OFMT) based on P2P lookup protocol in VDTNs, which can enable multimedia files to be sent to the receiver fast and reliably in wireless mobile P2P networks over VDTNs. In addition, a method of calculating the most suitable size of the fragment is provided, which is tested and verified in the simulation. And we also show that OFMT can defend a certain degree of DoS attack and senders can freely join and leave the wireless mobile P2P network. Simulation results demonstrate that the proposed scheme can significantly improve the performance of the file delivery rate and shorten the file delivery delay compared with the existing schemes.

1. Introduction

The applications of peer-to-peer (P2P) networks have been growing at tremendous speed these past few years. The P2P traffic was about 37.9% of the global Internet traffic [1]. P2P file sharing uses multiple peers to distribute contents, which can solve the bandwidth bottleneck highlighted by the C/S mode where multiclients download files from the same server simultaneously [2]. Therefore, multisource transmission is a popular architecture in P2P networks to increase the scalability and robustness. For large multimedia files, multisource transmission plays an irreplaceable role.

Although originally developed for the wired Internet, these P2P-based content distribution networks (also referred to as P2P-based file-sharing systems) now transcend network boundaries (wired or wireless). This is because a large number of wireless handheld devices introduced to the market in recent years have enhanced the trend of file distribution among the mobile users. Moreover, instead of the conventional cellular networks, low-cost wireless connectivity such as Bluetooth and IEEE 802.11 offers the mobile devices an alternative way to communicate with each other. By exploiting such low-cost wireless connectivity, MANETs, the automatically self-organized wireless networks without any preconfigured infrastructure, can be established to enable independent mobile users to interact with each other. Due to the common characteristics such as decentralized architecture, self-organization, and self-healing features between P2P networks and MANETs, this structure makes the P2P applications over MANETs feasible and popular [3–7].

Vehicular ad hoc networks (VANETs) are special MANETs in which the nodes are vehicles. In VANETs, vehicles establish temporary network connections and communicate with each other under self-organization. They can also

perform the distribution of data quickly and efficiently for the benefit of passengers' safety and comfort. However, there are many limitations preventing nodes communicating with others steadily. For example, the fast changing topology of vehicles, limited radio range, and so on, all result in that the amount of time nodes in radio range of one another is reduced. Accordingly, the duration of time that nodes are able to transfer data between one another as they pass is limited in VANETs. So, vehicle-based delay tolerant networks (VDTNs) are invited by all these issues. VANETs links are always intermittent and this interruption last longer, so that the path between the source node and destination node may not exist at any time; then VANETs become VDTNs [8]. VDTNs perform routing functions through store-and-forward mechanism [9–12], where a source node forwards messages to intermediate nodes moving into its transmission radio coverage. Meanwhile, these intermediate nodes store the received messages and forward them when an appropriate forwarding opportunity rises. Therefore, VDTNs enable nodes to be temporarily unreliable and long-standing disconnections. However, all of these characteristics affect the efficiency of P2P sharing. Therefore, the characteristic that at any time the path between the source node and destination node may not exist limits the P2P application over VDTNs. Fortunately, this paper presents a reasonable and effective solution.

Although some work has been done in the multi-source transmission of wireless mobile P2P networks [13–18], one of the key assumptions of the current solutions in the literature is the existence of end-to-end routing path between any two nodes. In this paper, we propose an optimal fragmentation-based multimedia transmission scheme (OFMT) based on P2P lookup protocol, which allows multimedia files to be transmitted to the receiver fast and reliably in the mobile P2P networks over VDTNs. In addition, there is no need for a centralized control point, and senders can freely join and leave from the wireless mobile P2P network. The distinct features of the proposed scheme include the following.

(1) It is common to split such big multimedia files into several fragments. However, the size of each fragment directly affects the efficiency of file transmission, for example, fragment delivery rate, transmission delay, network load, and so on. This point can be proved by the simulation results in Section 5. Therefore, we propose a method that the receiver calculates the size of multimedia files' fragments according to the real-time features of the networks.

(2) We present a multimedia transmission mechanism named optimal fragmentation-based multimedia transmission scheme (OFMT) based on P2P lookup protocol, which can be applied to VDTNs. In the meantime, it explores the full potential of each source nodes; thus the performance of multimedia file transmission achieves significant results including improving the fragment delivery rate and reducing the transmission delay.

(3) All that is needed is that the receiver calculates the size of fragment M in our OFMT scheme. So the algorithm complexity is $O(1)$.

(4) The node churn has small impact on the performance of OFMT.

The rest of the paper is organized as follows. In Section 2 we describe the related work about P2P lookup protocol and multi-source transmission in wireless mobile P2P networks; we present a distributed multi-source parallel coadjutant transmission method in Section 3. Section 4 provides the security analysis of our protocol. The simulation results are presented in Section 5. Section 6 concludes the paper.

2. Related Work

Multi-source transmission is a popular architecture in P2P networks to increase the scalability. A good example of multi-source architecture is the BitTorrent system [19]. In [5], a multi-source transmission protocol using network coding is presented. Multi-source real-time video transmission is studied in [13, 14, 16].

In a cooperative network with multiple potential relays and multiple simultaneous transmissions, the selection cooperation is presented in [15], wherein each source pairs with a single "best" relay. In [13], a peer-to-peer (P2P) service for the transmission of real-time video content is introduced, exploiting the contemporary usage of multiple network paths over the current Internet. Reference [16] presents a multi-source streaming approach to increase the robustness of real-time video transmission in MANETs. For that, video coding as well as channel coding techniques on the application layer is introduced. Reference [14] shows how video streaming applications can benefit from the diversity offered by P2P systems and implement distributed streaming and scheduling solutions with multipath packet transmission. An asynchronous multi-source streaming (AMSS) model [20, 21] is discussed to realize the scalable multimedia streaming service. Here, each of multiple sources sends only a part of a multimedia fragment to a receiver.

In [22], a heterogeneous asynchronous multi-source streaming (HAMS) model is discussed, where multiple sources transmit packets of a multimedia file to a requesting receiver to increase the throughput, reliability, and scalability in P2P overlay networks. The source nodes send fragments not in distributed manner, although parallel transmission mechanism is used in HAMS model. It needs to send some control packets among all the source nodes to determine which file fragments should be sent by each of them. Obviously, this model is only suitable for good network connectivity such as the Internet. It is impracticable that the control packets determine how to send fragments in VDTNs with high transmission delay and low message delivery rate. In the case of loss of control packets, the performance of this protocol almost cannot be guaranteed. In addition, the complexity of this model is high.

The literature [18] presents a multisource selection mechanism in MANETs. In this protocol, the time period of the multimedia transmission is divided into time slots. Each time

slot corresponds to one file fragment. In each time slot, it finds a source node with the best performance as the sender of this time slot and repeats the process until all the fragments are transmitted. Apparently, this method has low reliability. If the selected sender leaves the network or disconnects with the receiver, then the file cannot be transmitted normally. Therefore, it is only suitable for the networks with good connectivity. In [17], the author has improved the previous scheme by selecting M senders in each time slot to increase the reliability of the system. However, all the selected M senders still transmit fragments serially according to the time slots. That is, each time slot corresponds to one file fragment. As a result, the transmission delay is relatively high. Moreover, this protocol is only suited to the condition that the connection between the source node and the destination node exists. And it is not feasible in VDTNs.

All the multi-source transmission mechanisms discussed above only apply to the networks with good connectivity. In addition, most of them do not employ parallel idea to improve the throughput of network. Only [22] proposes a parallel transmission mechanism, but its application scenarios are limited. To address the problems and issues discussed above, it is desirable to develop a fast and reliable transmission scheme of the multimedia files which can apply to some special networks. Thus, in Section 3, we propose a distributed multi-source parallel coadjutant transmission of multimedia based on P2P lookup protocol in VDTNs.

3. OFMT Scheme Based on P2P Lookup Protocol

VDTN is a network model abstracted out of ad hoc, wireless sensor network (WSN) and other self-organizing wireless networks. Its typical characteristic is the link between nodes that is intermittently interrupted and usually the interruption lasts longer, so that at any time the path between the source node and destination node may not exist [8]. Therefore, the transmission mechanism of the multimedia files in the mobile wireless P2P networks over VDTNs is different in the current scheme.

In wireless mobile P2P networks, due to nodes joining and leaving the networks, the system performance may be dramatically affected. We call this phenomenon node churn [23, 24]. As a result, under the situation of unstable links, if a single source node is used, the churn of this source node will cause a sharp drop in the efficiency of files transmission. For this reason, in this work, we use multi-source transmission first. On the one hand, it increases the robustness of the system. On the other hand, it can provide multimedia file sharing at the same time and enhance the capacity and efficiency of real-time transmission [17]. Second, in our scheme, the fragments of a multimedia file are transmitted to a receiver from multiple sources in parallel to increase the throughput. Thirdly, multiple source nodes transmit their fragments in a fully distributed way. Finally, the proposed coadjutant transmission mechanism is implemented by letting the source nodes that have finished their own tasks automatically help the node that has sent the least fragments. It is worth emphasizing that the mechanism is still executed in a distributed manner.

Before the detailed description of the OFMT, we first introduce three types of nodes of the networks: the receiver node, the source node, and common intermediate node.

(i) The receiver node D, defined as the node that requests files.

(ii) The source node S, defined as the node storing the desired file found by the receiver executing a P2P lookup protocol.

(iii) The common intermediate node IN, defined as the nodes other than the receiver nodes and the source nodes, which are mainly responsible for storing and forwarding the messages.

3.1. Fragment Distribution to Multiple Sources. The intermittent connectivity of VDTNs means there may be no persistent existence of connections between the source node and the destination node. That is, the links among nodes are very unstable and the duration of connections is very short. During packet transmission, if the link is down then the part of the fragment that has already been sent will be dropped. Therefore, to improve the fragment delivery rate, it is necessary to divide such big multimedia files into fragments with suitable size based on the actual link condition of the networks. Meanwhile, these fragments must be indexed.

In this paper, we propose to determine the fragment size according to the mean duration of the network links.

First, the receiver node D collects the historical information of the networks including the number of network connections C and the total duration of network connections t.

Second, before sending the fragment request, the receiver node D calculates the mean duration of the network links that equals t/C and then calculates the size SM of total packets transmitted during the lifetime of a link using the nodes' average transmission speed (bandwidth). However, in the best cases, that is to divide SM into several packets in VDTNs. In our experiments, we found that fragment's size of SM/5 allows for reasonable results in a variety of networks.

Third, in our method, the task list Stask[] of each source node S is the fragments with id $L, L+N, L+2N, \ldots, Q$, where L is the ordinal of S in the address list of the source nodes searched by D, N is the number of the source nodes found by D, and Q is the largest integer satisfying$Q = L + iN \leq$ the biggest fragment id ($i = 0, 1, 2 \ldots$).

As described above, each node is assigned as many tasks as each other. We adopt coadjutant transmission mechanism in OFMT; thus, the good-performance nodes will finish tasks earlier than the nodes with poor-performance, and then those nodes with earlier completion help the unfinished nodes until the entire file is transmitted. Therefore, our scheme takes into account nodes with various properties while not reducing the throughput of the system.

3.2. Data Structure. Here, we set the size of the entire multimedia file as f Size, the number of source nodes as N, the fragment size as M, the address list of source nodes as

Saddrs = $[P_1, P_2, P_3, \ldots, P_N]$ (suppose the ith source node address searched by the receiver is P_i, $i = 1, 2, 3 \ldots N$), the fragment ID as $f_{\mathrm{Id}} = 0, 1, \ldots Q$, Q = Math.ceil(f Size/M), and the requested file ID as $\mathrm{F_{Id}}$.

3.2.1. ACK. In OFMT, all the nodes of the networks need to maintain a global success list ACK$[0 \cdots N-1][0 \cdots f\mathrm{Num}-1]$ to store the fragment IDs that have been sent to the receiver D successfully, where f Num = Math.ceil(f Size/$(M * N)$). ACK$[i][j] = i + j * N$, $(i = 0, 1, \ldots, N-1; j = 0, 1, \ldots, N-1)$ when the node $P(i+1)$ has sent its $(j+1)$th fragment and the receiver D has received it successfully.

Once the receiver D successfully receives a file fragment (fragment ID is f_{Id}), it checks ACK[][] to see if ACK$[s][f]$ exists, where $s = f_{\mathrm{Id}} \% N$, $f = f_{\mathrm{Id}}/N$. If ACK$[s][f]$ exists, it means the fragment f_{Id} has already been received and then this fragment will be dropped. On the contrary, if ACK$[s][f]$ does not exist, it means the receiver D has not received this fragment before and starts to receive it immediately. At the same time, set ACK$[s][f] = f_{\mathrm{Id}}$. By using this data structure, a fragment id f_{Id} can be directly targeted to its storage location ACK$[s][f]$ in ACK[][] by any nodes, where $s = f_{\mathrm{Id}} \% N$, $f = f_{\mathrm{Id}}/N$. As a result, this data structure reduces the algorithm complexity.

3.2.2. Message. In our scheme, three types of messages are introduced in the multi-source transmission process.

(i) Initial-multicast-notification-request (IMIR) message is a multicast-notification message sent to all the source nodes from the receiver after D executes a P2P lookup protocol to find the desired file. IMIR mainly includes the address of D, the address list SAddrs of N source nodes, the requested file id $\mathrm{F_{Id}}$, and the fragment size M.

(ii) Unicast-notification-request (UIR) message is a unicast-notification message sent to the node A from D when D checks ACK[][] and finds there is a node A sending none of the fragments. UIR mainly includes the address of D, the address of the node A, the requested file id $\mathrm{F_{Id}}$, and the fragment size M.

(iii) Source-fragment (SF) message is a fragment-message sent to the receiver D from a source node S after this source node receives a notification message from D. SF mainly includes the address of the source node, the address of D, the requested file id $\mathrm{F_{Id}}$, the fragment id f_{Id}, and the relevant multimedia file content.

3.2.3. Timeout Timer. In our scheme, each node needs to maintain a timeout timer.

(i) The timeout timer at S: each source node S starts the timer after sending an SF message. When the timer expires, S starts to send the next file fragment until all fragments have been successfully sent.

(ii) The timeout timer at D: D starts the timer each time it sends an IMIR or UIR message. When the timer expires, it checks ACK[][]. If there is an uninformed source node, then D sends UIR message to this node. This process is repeated until all the source nodes are notified or the number of the timer expiring exceeds a certain threshold.

(iii) The timeout timer at IN: each intermediate node IN starts the timer at the beginning of transmitting the file. When the timer expires, they check whether the current carried messages are successfully transmitted.

3.3. OFMT Scheme. With the above data structure, OFMT scheme is described below.

(1) The receiver D executes a P2P lookup protocol to find the desired file, and then the address list of the source nodes is sent to the receiver. Suppose there are N source nodes.

(2) The receiver node D collects the historical information of the networks to calculate the average duration of the network links and then calculates the size SM of total packets transmitted during the lifetime of a link based on the nodes' average transmission speed (bandwidth). In our experiments, we found that fragment's size of SM/5 allows for reasonable results in a variety of networks.

(3) D sends IMIR message to N source nodes using multicast to inform them to divide the multimedia file $\mathrm{F_{Id}}$ into fragments with size M and requests the first fragment. In the meantime, D starts its timeout timer.

(4) After receiving IMIR or UIR message, each node in the networks compares its address with the destination address list of the message:

> if this node is a common intermediate node, then it continues to forward this message.

If this node is one of N source nodes, then it divides the file $\mathrm{F_{Id}}$ into fragments with size M and sends the first fragment of its task list Stask[]; meantime, it starts its timeout timer after sending the fragment.

(5) After receiving a fragment successfully, the receiver D puts its fragment id into ACK[][], which means that this fragment has been successfully transmitted.

(6) When the timeout timer expires, do the following actions until all the fragments have been transmitted successfully.

(i) The receiver D: check ACK[][] whether the fragments whose ids are from 0 to $N - 1$ have been successfully sent.

(1) If any fragment with id ranging from 0 to $N - 1$ is missing in ACK[][] and the number of timer expirations exceeds 3, then either the relevant source node is dead or has left the networks. Therefore, D no longer sends notification messages and waits for the assistance of the other nodes and removes the timer.

(2) If any fragment with id ranging from 0 to $N-1$ is missing in ACK[][] and the number of timer expirations is less than 3, then D sends UIR to the relevant source nodes.

(3) If the fragments whose ids are from 0 to $N-1$ have all been sent to D, then D removes the timer, which means that then N source nodes have already been notified and the next step is to rely on them to complete fragments transmission in a distributed manner.

(ii) The source node S: here, we suppose the prior file fragment id sent by S is I.

(1) If S does not finish the transmission of the fragments in S's task list Stask[].

If $(I + N) \leq$ Math.ceil(f Size/M), S sends the fragment with id $I + N$.
If $(I + N) >$ Math.ceil(f Size/M), it means S has already sent all the fragments of its task list, then go to (2).

(2) If S has already sent all the fragments of Stask[], then run the coadjutant algorithm (Algorithm 1).

(iii) The common intermediate node IN checks ACK[][]. If a current-carried fragment has been sent successfully to the receiver D, then IN removes this fragment and stops forwarding. If on the contrary, IN continues to forward this fragment to the next hop according to DTNs' routing protocols.

From the above steps, it can be drawn that even low-performance and low-reliability mobile nodes can also be a sender in our scheme. This is due to the coadjutant mechanism, which makes good-performance nodes account for more fragments and poor-performance nodes send fragments as possible as it can. Therefore, the throughput of the system is not affected. In addition, our OFMT scheme tries best to reduce the redundant transmission of packets and increase the reliability and availability. We also have the highest file delivery rate and the lowest transmission delay.

4. Safety Analysis of OFMT Scheme

4.1. Denial of Service (DoS). The transmission of the multimedia file depends on the cooperation of each source node. Therefore, one of the most worrisome results is the possibility of a denial of service (DoS) attack where malicious nodes refuse to transfer file fragments to the requesting node. Selfish nodes performing this attack attempt to benefit from the resources of others without offering their own resources in exchange [25]. The selfish nodes attempt to stop, or at least slow, file delivery rate [26].

In this case, the performance is equivalent to the cases that fewer senders are selected or the selected senders leave the network. However, as a result of using coadjutant mechanism in our scheme, even in such circumstances that if not all source nodes are selfish nodes, these surviving nodes still can finish the transmission of the files while system performance does not fall a lot, which can be seen from Figures 7 and 9.

4.2. Impacts of Nodes Churn. In OFMT, all nodes can be divided into three categories: source nodes, common intermediate nodes, and the receiver nodes. Therefore, 6 cases are presented to discuss the effect on system performance due to node churn.

(1) The case of a source node leaving the networks: if the leaving node has good performance, the system is like losing a right-hand man, resulting in increased delay. However, the system can still rely on other nodes to complete the transmission of all fragments. In contrast, if a poor-performance node leaves the network, it can mandate other good-performance nodes to help to complete the transmission. Meanwhile, the system performance is not significantly decreased.

(2) The case of a source node joining the networks: if a source node rejoins the network after it left the network, and then at the time when it rejoins, it can continue its task. Thus the completion of the file transmission can be accelerated.

(3) The case of the common intermediate nodes leaving the networks: this case makes VDTNs nodes sparser and often leads to network fragmentation. Therefore, it will have an impact on VDTNs routing such as reducing the message delivery rate. The system performance is also affected.

(4) The case of the common intermediate nodes joining the networks: VDTNs nodes are denser in this case. Therefore, the message delivery rate of VDTNs routing is improved as well as the system performance.

(5) The case of the receiver node leaving the networks: in this case, because of the absence of the receiver, it does not matter whether the network performance is good or bad. This is a usual case in VDTNs.

(6) The case of the receiver node joining the networks: if the receiver node rejoins the network after it left the network, the system is back to normal. The source nodes continue to transmit the file fragments and the system performance will not be affected.

5. Simulation Results and Discussions

To begin this section, we simply introduce the existing algorithm multi-source serial transmission (MST) based on time slot. In MST, the time period of the multimedia transmission is divided into time slots. Within each time slot, all source nodes or only the good-performance nodes send the same file fragment. Moreover, in this algorithm, the fragment size totally depends on the receiver, not the actual situation of the networks. In this section, we simulate OFMT and MST. The transmission structures of OFMT and MST are shown in Figures 1 and 2.

```
if (All the fragments of Stask[] have been successfully transmitted to the receiver D)
{
Find the source node minS that has sent the least fragments in ACK[][]. Check the task
list of minS reversely and find the first fragment that isn't sent successfully. Then S helps
min S to send this fragment.
}
else
{
Check the task list Stask[] sequentially in ACK[][] and find the first fragment that isn't
sent successfully. Then, S sends this fragment.
}
```

Algorithm 1: Coadjutant algorithm.

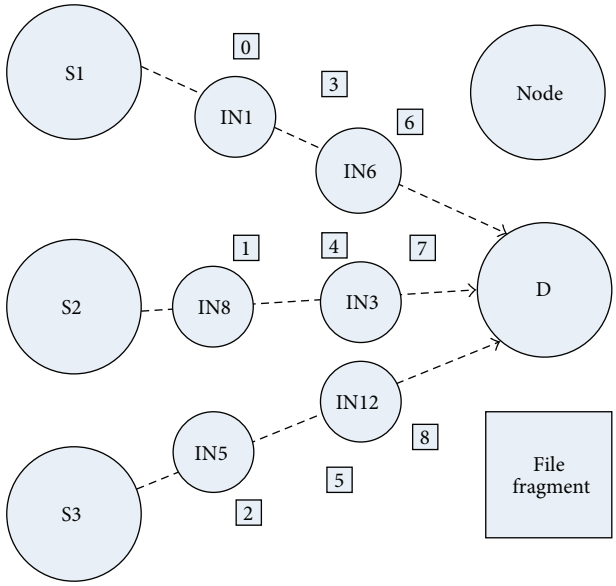


Figure 1: The transmission structure of OFMT.

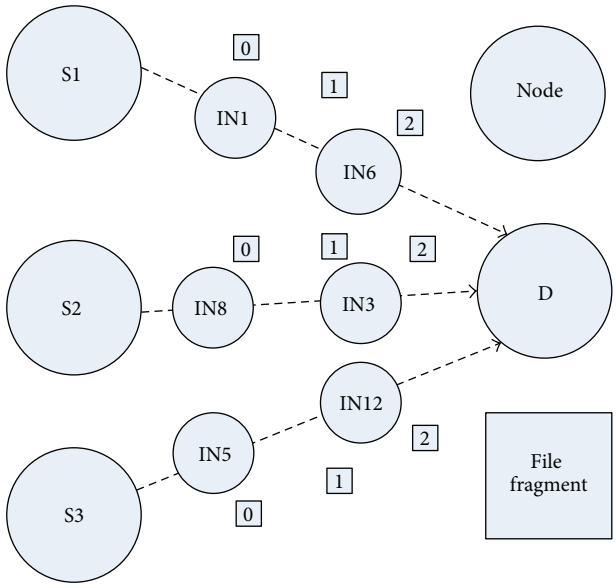


Figure 2: The transmission structure of MST.

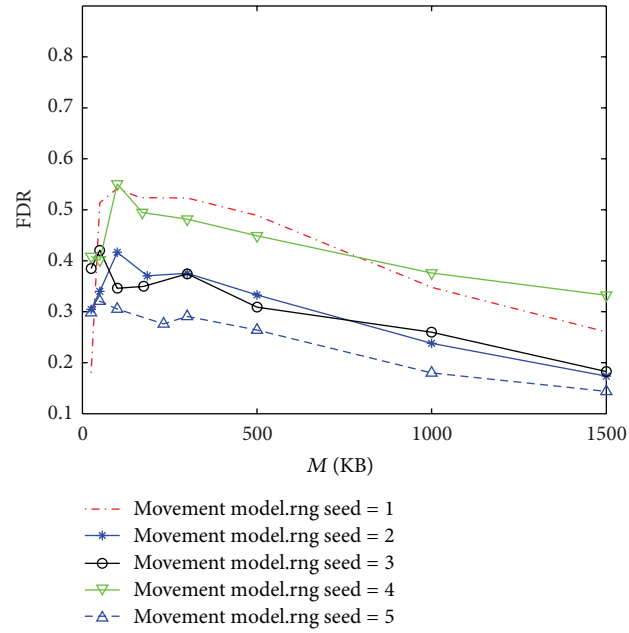


Figure 3: The relationship between file delivery rate and fragment size M (simulation time = 60 Ks, N = 10, transmitting range = 10 m).

5.1. Simulation Settings. In our simulation, we use extended MChord [24] P2P lookup protocol, and the transmission protocol is simulated by the opportunistic network environment simulator (ONE) [27, 28]. We assume interpersonal communication between mobile users in a city using modern mobile phones or similar devices, using Bluetooth at 2 Mbit/s net data rate with 10 m radio range [29]. The mobile devices have up to 100 MB of free buffer space for storing and forwarding messages.

Therefore, based on the above scenario, the simulation parameters are set as in Table 1. There are three types of nodes in the networks which are used to simulate the movement of pedestrians, cars, and buses. All trajectories are based on the Helsinki map.

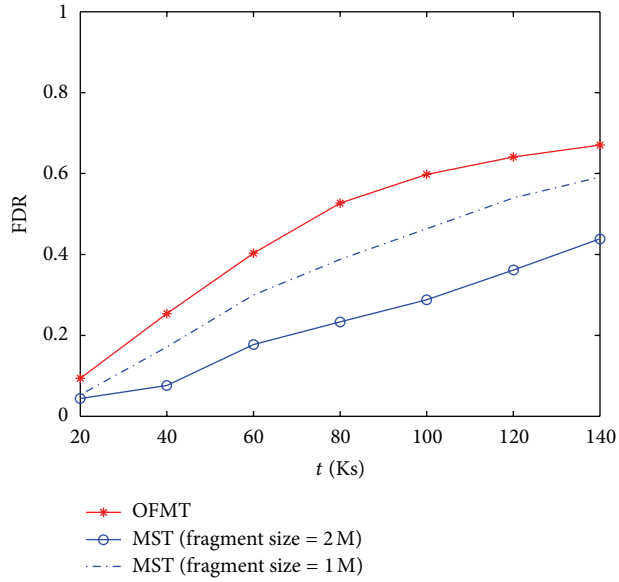


Figure 4: The relationship between file delivery rate and transmission time ($N = 10$, transmitting range = 10 m).

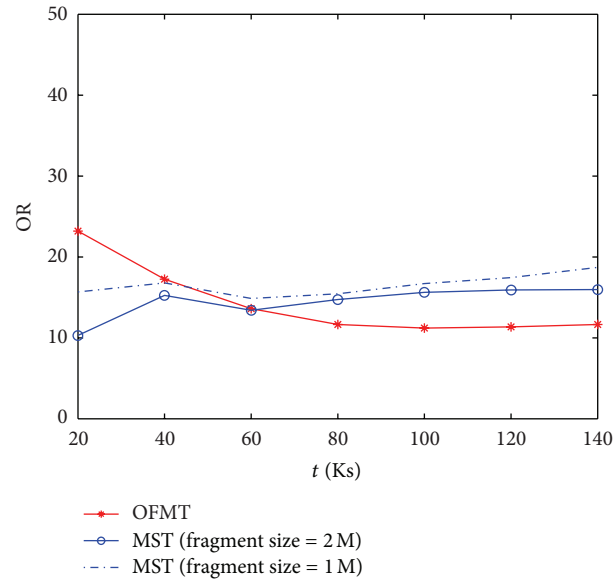


Figure 5: The relationship between overhead ratio and transmission time ($N = 10$, transmitting range = 10 m).

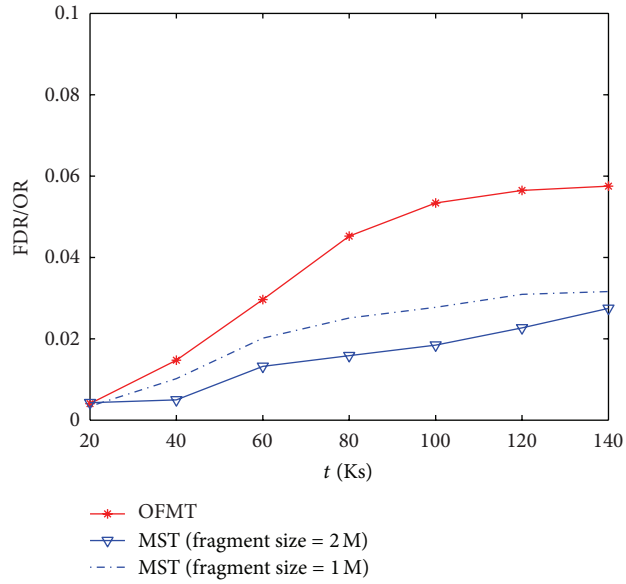


Figure 6: The relationship between file delivery rate/overhead ratio and transmission time ($N = 10$, transmitting range = 10 m).

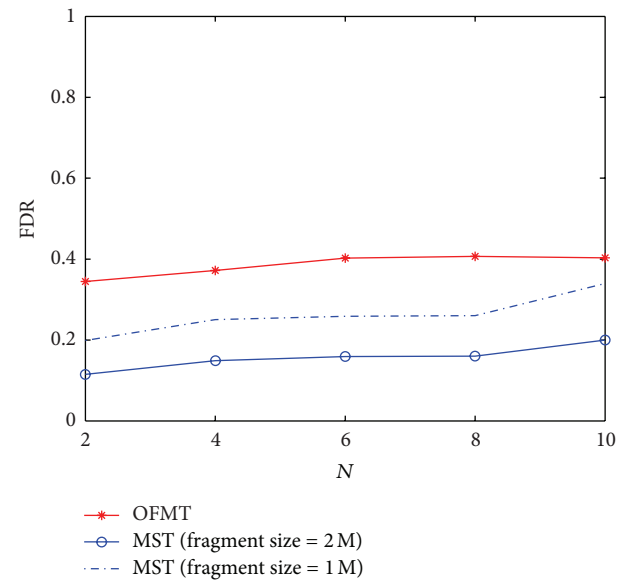


Figure 7: The relationship between file delivery rate and the number of source nodes (simulation time = 60 Ks, transmitting range = 10 m).

In the following simulation, we compare OFMT with two cases of MST: MST (Fragment size = 1 M) and MST (fragment size = 2 M).

5.2. Simulation Results. In Figure 3, we show the relationship between file delivery rate (FDR) and fragment size M under 5 different movements. As can be seen from the figure, the fragment size can greatly affect FDR in the networks. We find that when the seed of the movement model is set to 1, 2, and 4, the corresponding curves can obtain the maximum at the third point where the fragment size is 100 KB and the curves whose movement seed is 3, 5 can obtain the maximum at the second point where the fragment size is 50 KB, while the fourth point where the fragment size is about 150 KB in every curves is found by our OFMT scheme (M = the average duration of the network links/5). When the file size is a multiple of megabyte or even gigabyte, the difference between the value found by our OFMT scheme and the value where FDR is the highest is less than 100 KB. Meanwhile, the corresponding FDR of the two differs within 0.05. Obviously, our method can get the approximate optimal value of M to

Table 1: Simulation parameters.

Parameter	Value
Buffer size	100 M, 500 M (buffer size of the receiver)
Transmit range	10 m, 50 m, 100 m
Transmit speed	250 kBps
Wait time	0, 120
Speed	(2.7, 13.9) (7, 10)
nrof hosts	44
Message sizes	M (the calculated fragment size), 1000 KB, 2000 KB
World size	5000, 4300
Router	Spray and wait router
Movement model	Map route movement (Helsinki map)
Number of senders	2, 4, 6, 8, 10
File size	500 M

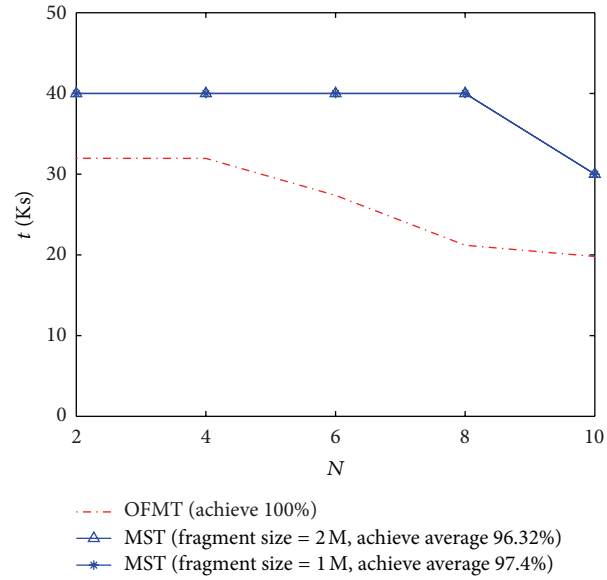


Figure 8: The relationship between the time of achieving the max FDR and the number of source nodes N.

get a much larger FDR. Therefore, Figure 3 shows that our proposed method provides an effective way to look for the value of M. In the following simulation, the value of M in OFMT scheme equals the average duration of the network links/5.

Figure 4 shows the changing tendency of FDR over time both in OFMT and MST. We can see that the FDR of both algorithms is increasing over time, and the FDR in OFMT is much higher than two cases of MST. One reason for this is that OFMT scheme can calculate the fragment size M according to the actual situation of the networks. Another is due to OFMT scheme using parallel coadjutant transmission. As a result, on the one hand, parallel transmission can improve the throughput of the system. On the other hand, coadjutant transmission can solve the problem that the message delivery rate is low in VDTNs; that is, it helps to resend those dropped packets due to disconnected links. Therefore, OFMT scheme improves FDR.

We plot the overhead ratio (OR) against the time in Figure 5. The OR is defined as the ratio of the number of the messages that failed to reach the destination and the messages that are transmitted to the destination successfully. The larger the OR is, the more overhead is required in every transmission of a packet. As can be seen from the figure, the changing tendency of OR over time is not obvious, and the OR of the three is very comparable. Before 60 Ks, OFMT has a larger OR than MST. However, after 60 Ks, the situation is exactly opposite. Therefore, we consider a composite metric FDR/OR which takes FDR and penalizes it for having a poor OR. So we maintain the standard of "higher is better." It is very clear that the performance of OFMT is better than MST from Figure 6. At the same time, it also demonstrates that OFMT can achieve a higher FDR while maintaining a lower OR.

Figure 7 provides the relationship between FDR and the number of the source nodes. We can see that the FDR of OFMT is always higher than both two cases of MST. Meanwhile, the FDR of both OFMT and MST does not change much while the number of the source nodes ranges from 2 to 10, which illustrates that the source node churn have small impact on the performance of networks.

Figure 8 shows the relationship between the number of the source nodes and the time of both schemes achieving the largest FDR when the transmitting range is 100 m. Two curves of MST whose fragment sizes are 2 M and 1 M, respectively, overlap. Measured by our experiments, the largest FDR of OFMT is 100%. However, two cases of MST can only achieve 96.32% and 97.4%, respectively. This is because MST transmits the packets based on time slots and each time slot corresponds to one file fragment. However, VDTNs can neither guarantee the message delivery time nor guarantee that each fragment in each time slot be successfully transmitted to the receiver. In addition, from the figure, we can see that OFMT achieves the highest FDR earlier than MST. This means OFMT scheme has a much lower transmission delay.

We plot the FDR of both OFMT and MST (fragment size = 1 M) against time with different transmitting ranges in Figure 9. It is clear that the FDR of both algorithms increases with time under any transmitting range. What is more, OFMT's FDR is higher than that of MST. In addition, we can also observe that the larger the transmission range is, the higher the FDR is. This is because the larger transmission range results in better network connectivity and higher message delivery rate, and ultimately higher FDR.

6. Conclusions

This paper proposes an optimal fragmentation-based multimedia transmission scheme (OFMT) based on P2P lookup protocol in VDTNs. More specifically, three mechanisms enable the highest file delivery rate and the lowest transmission delay. In addition, we design a method to

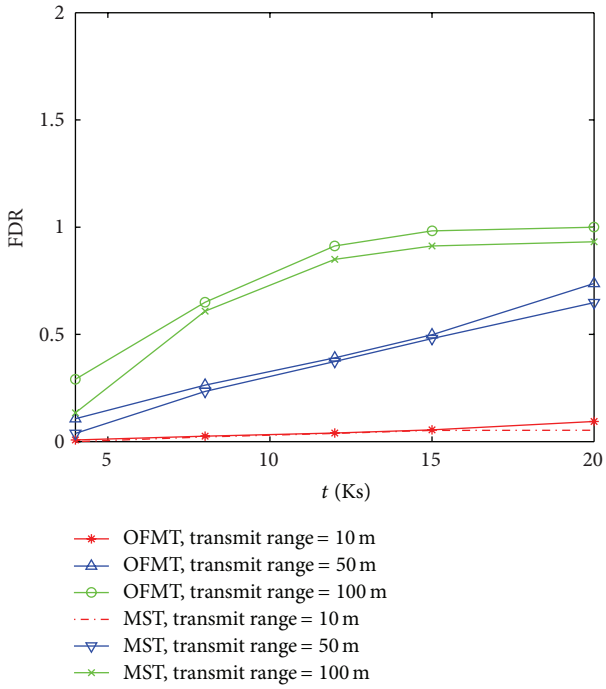


Figure 9: The relationship between file delivery rate and transmission time ($N = 10$).

get a best suitable fragment size which allows for reasonable results in a variety of networks. Moreover, we show the safety analysis of OFMT scheme including DoS attack and node churn. Finally, the results of simulation demonstrate that the proposed scheme significantly improves the performance of file delivery rate and file transmission delay compared with the existing scheme. Therefore, OFMT provides higher level of securities.

Acknowledgment

This work is supported by the Open Research Fund from the Key laboratory for Computer Network and Information Integration (Southeast University, Ministry of Education, China), the Fundamental Research Funds for the Central Universities, National Key Technology R&D Program(Grant no. 2011BAK02B02-01), High-Tech 863 Program (no. 2012AA111902), National Key Technology R&D Program of China (no. 2011BAK02B02), State Key Development Program for Basic Research of China (Grant no. 2011CB302902), National Natural Science Foundation of China (Grant no. 61073180), National Science and Technology Major Project (Grant no. 2010ZX03006-002-03), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, the National Natural Science Foundation of China (Grant no. 60933011), and the National Science and Technology Major Project (Grant no. 2010ZX03006-004).

References

[1] "Global IP Traffic Forecast and Methodology, 2006–2011," Cisco Syst. White Paper, January 2008.

[2] J. Qian and H. Yang, "The ideas of P2P file sharing," *Network Communications and Security*, pp. 955–956, 2007 (Chinese).

[3] A. Duran and C. C. Shen, "Mobile ad hoc P2P file sharing," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 1, pp. 114–119, Atlanta, Ga, USA, March 2004.

[4] S. Rajagopalan and C. C. Shen, "A cross-layer decentralized BitTorrent for mobile ad hoc networks," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous '06)*, pp. 1–10, SanJose, Calif, USA, July 2006.

[5] C. C. Hoh and R. H. Hwang, "P2P file sharing system over MANET based on swarm intelligence: a cross-layer design," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 2674–2679, Hong Kong, March 2007.

[6] B. Tang, Z. Zhou, A. Kashyap, and T. C. Chiueh, "An integrated approach for P2P file sharing on multi-hop wireless networks," in *Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '2005)*, pp. 268–274, Montreal, Canada, August 2005.

[7] H. Pucha, S. M. Das, and Y. Charlie Hu, "Ekta: an efficient DHT substrate for distributed applications in mobile ad hoc networks," in *Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '04)*, pp. 163–173, Lake District National Park, UK, December 2004.

[8] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*, pp. 27–34, Karlaruhe, Germany, August 2003.

[9] M. C. Chuah, P. Yang, B. D. Davison, and L. Cheng, "Store-and-forward performance in a DTN," in *Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC '06)*, pp. 187–191, July 2006.

[10] F. Hou and X. Shen, "An adaptive forwarding scheme for message delivery over delay tolerant networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–5, December 2009.

[11] R. Wang, X. Wu, T. Wang, and T. Taleb, "Experimental evaluation of Delay Tolerant Networking (DTN) protocols for long-delay cislunar communications," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–5, December 2009.

[12] H. Guo, J. Li, and Y. Qian, "HoP-DTN: modeling and evaluation of homing-pigeon-based delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 857–868, 2010.

[13] A. De Mauro, D. Schonfeld, and C. Casetti, "A peer-to-peer overlay network for real time video communication using multiple paths," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 921–924, Toronto, Canada, July 2006.

[14] D. Jurca, J. Chakareski, J. P. Wagner, and P. Frossard, "Enabling adaptive video streaming in P2P systems [Peer-to-peer multimedia streaming]," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 108–114, 2007.

[15] E. Beres and R. Adve, "Selection cooperation in multi-source cooperative networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 118–127, 2008.

[16] T. Schierl, K. Gänger, C. Hellge, T. Wiegand, and T. Stockhammer, "SVC-based multisource streaming for robust video transmission in mobile ad hoc networks," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 96–103, 2006.

[17] P. Si, F. R. Yu, H. Ji, and V. C. M. Leung, "Distributed multisource transmission in wireless mobile peer-to-peer networks: a restless-bandit approach," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 420–430, 2010.

[18] P. Si, F. R. Yu, H. Ji, and V. C. M. Leung, "Distributed sender scheduling for multimedia transmission in wireless mobile peer-to-peer networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4594–4603, 2009.

[19] B. Choen, "Incentives build robustness in BitTorrent," in *Proceedings of the P2P Economics Workshop*, pp. 978–982, Berkeley, Calif, USA, June 2003.

[20] S. Itaya, T. Enokido, and M. Takizawa, "A high-performance multimedia streaming model on multi-source streaming approach in peer-to-peer networks," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05)*, vol. 1, pp. 27–32, March 2005.

[21] S. Itaya, T. Enokido, A. Yamada, and M. Takizawa, "A scalable multimedia streaming model based-on multi-source streaming concept," in *Proceedings of the 11th International Conference on Parallel and Distributed Systems Workshops (ICPADS '05)*, vol. 1, pp. 15–21, July 2005.

[22] S. Itaya, N. Hayashibara, T. Enokido, and M. Takizawa, "Scalable peer-to-peer multimedia streaming model in heterogeneous networks," in *Proceedings of the 7th IEEE International Symposium on Multimedia (ISM '05)*, pp. 208–215, Irvine, Calif, USA, December 2005.

[23] M. Thiele, K. Kutzner, and T. Fuhrmann, "Churn resistant de Bruijn networks for wireless on demand systems," in *Proceedings of the 3rd Annual Conference on Wireless on Demand Network Systems and Services*, pp. 154–158, Les Ménuires, France, January 2006.

[24] G. Xu, W. Ma, G. Wang, X. Liu, and J. Liu, "Churn impact on replicated data duration in structured P2P networks," in *Proceedings of the 9th International Conference on Web-Age Information Management (WAIM '08)*, pp. 455–460, Zhangjiajie, China, July 2008.

[25] M. Feldman and J. Chuang, "Overcoming free-riding behavior in peerto- peer systems," *ACM SIGecom Exchanges*, vol. 5, no. 4, pp. 41–50, 2005.

[26] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 846–854, April 2009.

[27] A. Keränen and J. Ott, "Increasing reality for DTN protocol simulations," Tech. Rep., Helsinki University of Technology, Networking Laboratory, July 2007.

[28] ONE simulator, http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[29] A. Keränen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09)*, p. 55.